



# Ingeniería Web: Visión General

-IWVG-

# DevOps

Jesús Bernal Bermúdez

# ¿Qué es DevOps?

## Development – Operations & QA

Prácticas para unificar el desarrollo de software y la operación de negocio

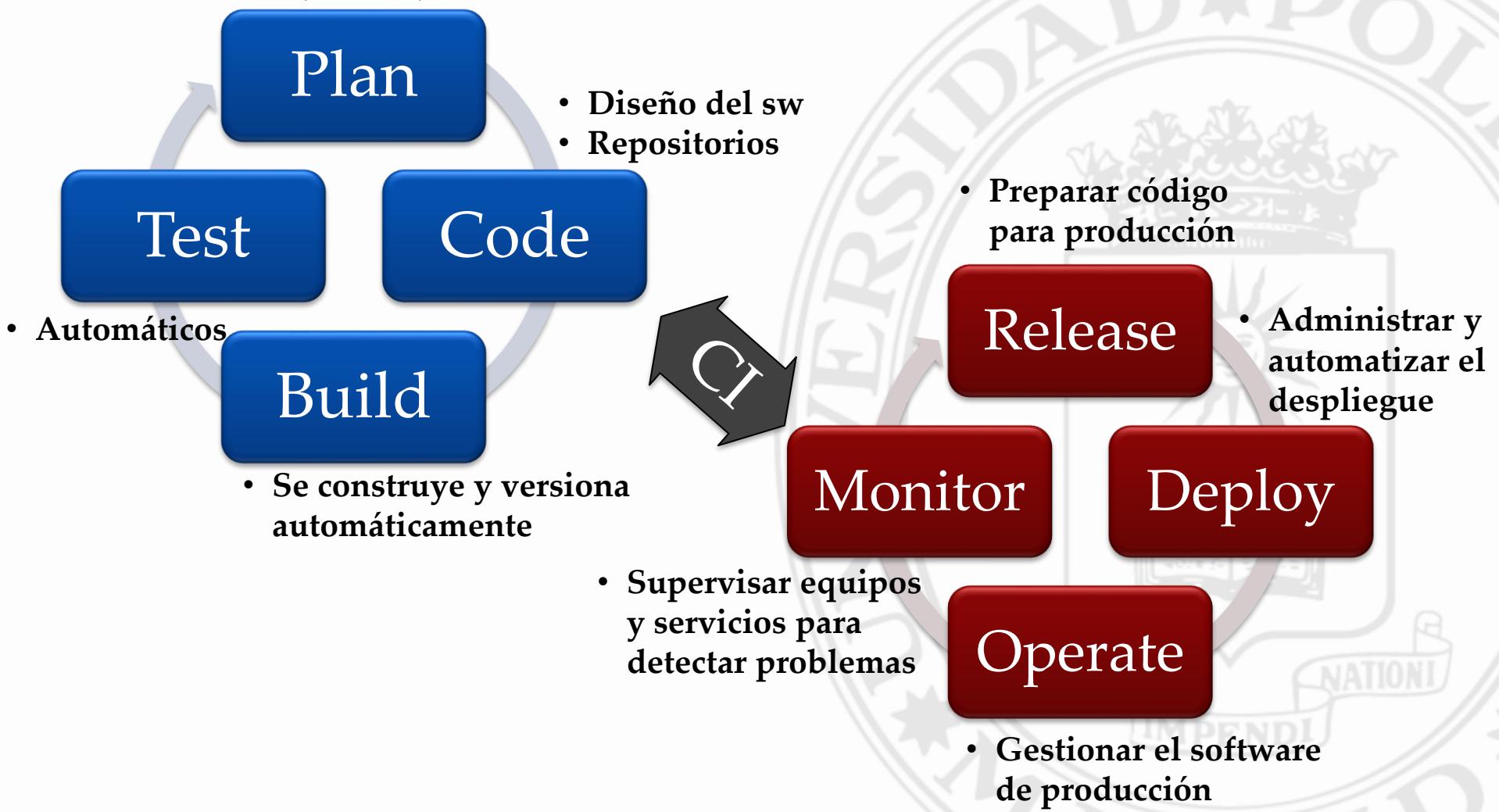
El objetivo es integrar los procesos y las herramientas para optimizar la entrega continua de software de calidad, siguiendo los objetivos empresariales

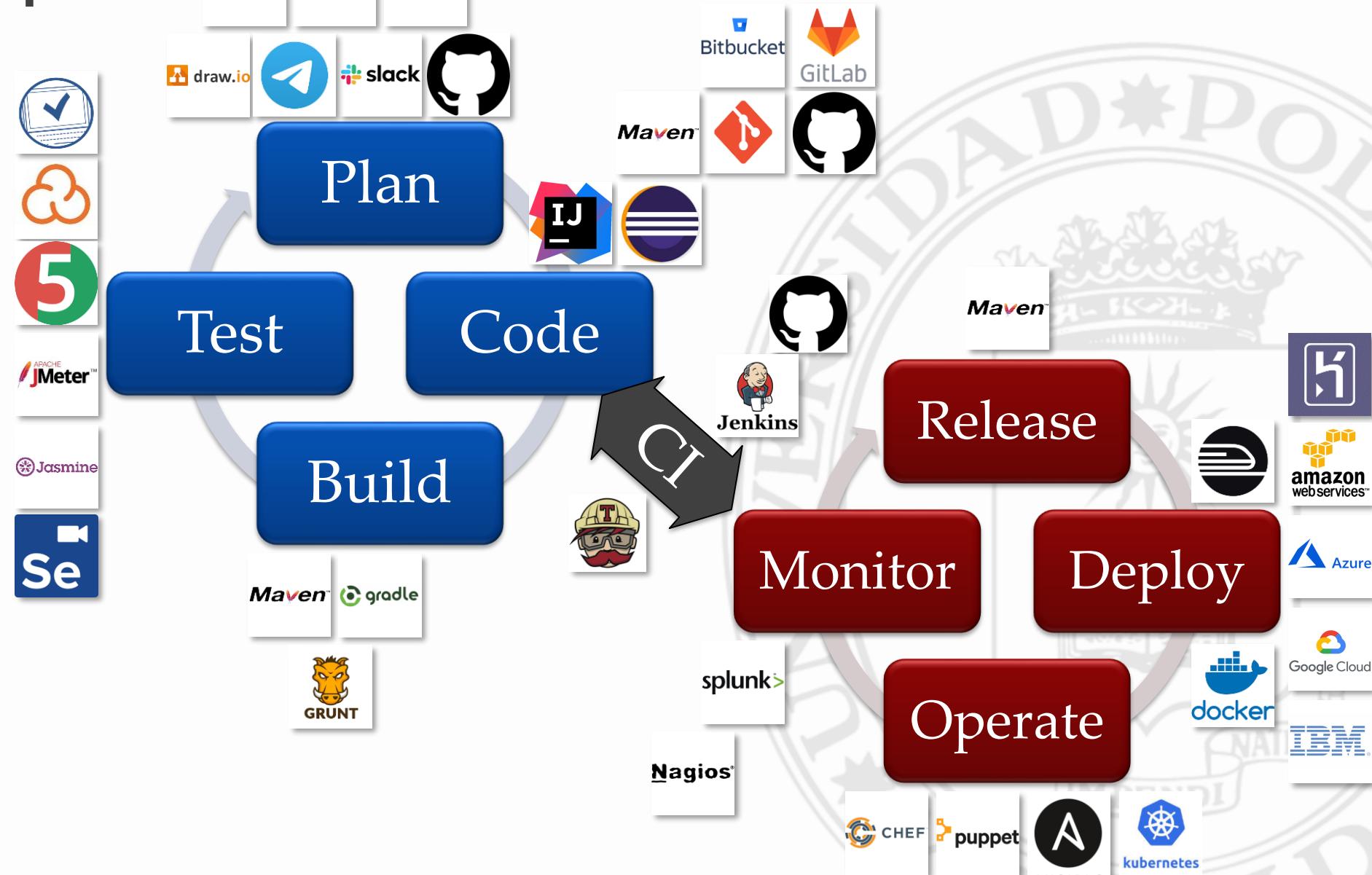
# ¿Qué es un Ecosistema Software?

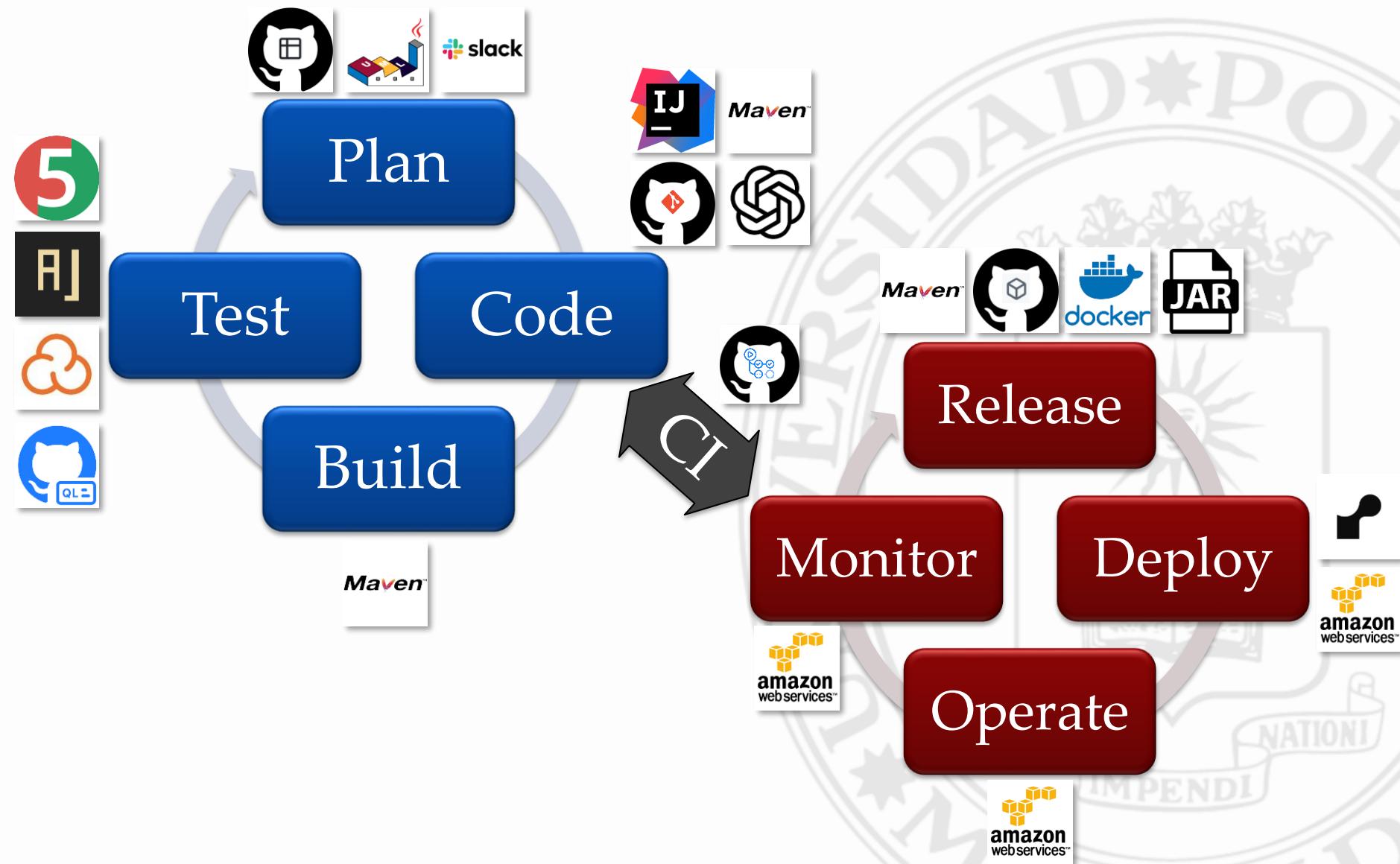
## Ecosistema Software

Se define como un espacio de trabajo donde un conjunto de herramientas interactúan y funcionan como una unidad para la colaboración, desarrollo, despliegue y supervisión del software en todas sus fases

- Valor comercial
- Gestión del proyecto
- XP, Scrum, Kanban...







# Maven Conceptos

## Apache Maven

Es una herramienta de gestión y construcción de proyectos de software con Java

pom.xml

## Artefacto

Componente software

Unidad mínima con la que trabaja Maven

## Coordenadas

Identificador único artefactos en Internet

### Group Id

- Identificación del grupo. Dominio al revés:  
*es.upm.miw*

### Artifact Id

- Identificación del artefacto: *devops*

### Version

- *1.0.0-SNAPSHOT*
- *1.3.4-RC (BETA...)*
- *1.4.5-Release*

## Empaquetado

Tipo de artefacto

JAR, POM

WAR, EAR,  
RAR...

# Maven Comandos

**clean**

- Elimina los ficheros generados en construcciones anteriores.

**validate, compile, test (\*.test)**

- Valida, compila y lanza tests unitarios.

**package**

- Genera el empaquetado final (**jar**, war...).

**integration-test**

- Despliega el paquete y ejecuta los test de integración (\*IT) & (\*FT).

**verify**

- Verificar que el paquete cumpla los criterios de calidad.

**install, deploy**

- Instala el paquete en el equipo local, deploy en repositorio remoto.

## Plugin

- Es una tarea específica, más pequeña que una fase de construcción, que contribuye a la construcción y gestión del proyecto.

**jacoco**

- Genera un informe de cobertura de los test.

**sonar:sonar**

- Se conecta con *Sonarcloud* para inspeccionar el código.

**spring-boot:run**

- Arranca una aplicación en local realizada con *Spring Boot*.

# Maven POM.XML

m pom.xml (iwvg-devops) ×

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <project
3      xmlns="http://maven.apache.org/POM/4.0.0"
4      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5      xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
6      <modelVersion>4.0.0</modelVersion>
7
8      <parent>
9          <groupId>org.springframework.boot</groupId>
10         <artifactId>spring-boot-starter-parent</artifactId>
11         <version>3.4.2</version>
12         <relativePath/>
13     </parent>
14
15     <groupId>es.upm.miw</groupId>
16     <artifactId>iwvg-devops</artifactId>
17     <version>2.1.0-SNAPSHOT</version>
18     <packaging>jar</packaging>
19     <name>${project.groupId}.${project.artifactId}</name>
20     <url>http://github.com/miw-upm/${project.artifactId}</url>
21
22     <licenses...>
23
24     <developers...>
25
26     <properties...>
27
28     <dependencies...>
29         <build>
30             <plugins...>
31                 </build>
32             </project>
```

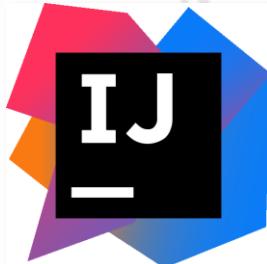
# Maven IDE

## *IntelliJ IDEA (JetBrains)*

- IDE para Java
- Tiene una versión gratuita: *Community Edition*.

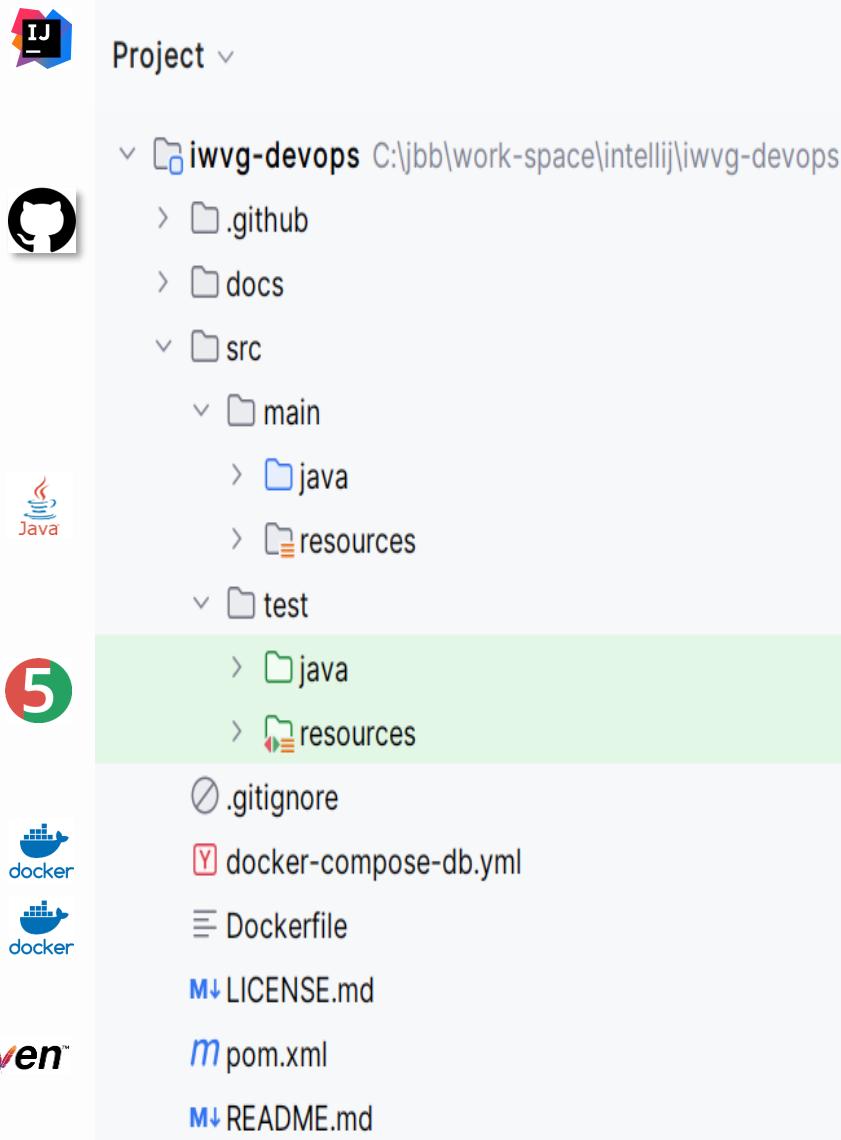
## Crear un proyecto nuevo

- Partir de un *Template*
- <https://github.com/miw-upm/iwvg-devops/docs>
  - Descomprimirla y cambiar coordenadas
- *Open desde IntelliJ*



# Maven

## Estructura del proyecto



<https://openjdk.org/>

**Versión: 21+**

**Variables de entorno:**

- **JAVA\_HOME:** Al raíz.
- **PATH:** Al bin

# Programación Orientada a Objetos

# Programación Funcional

## POO

Programación imperativa:  
¿**Cómo**?

Abstracción, encapsulamiento,  
modularidad, jerarquía.

Clases relacionadas mediante  
herencia, composición, agregación  
y asociación.

Objetos con estado: los atributos.

### Modelos

## PF

Programación declarativa: ¿**Qué**?

Basado en Funciones: funciones  
Lambda.

Sin estado, sin orden y sin efectos  
colaterales.

Valores inmutables: paso de  
parámetros por valor.

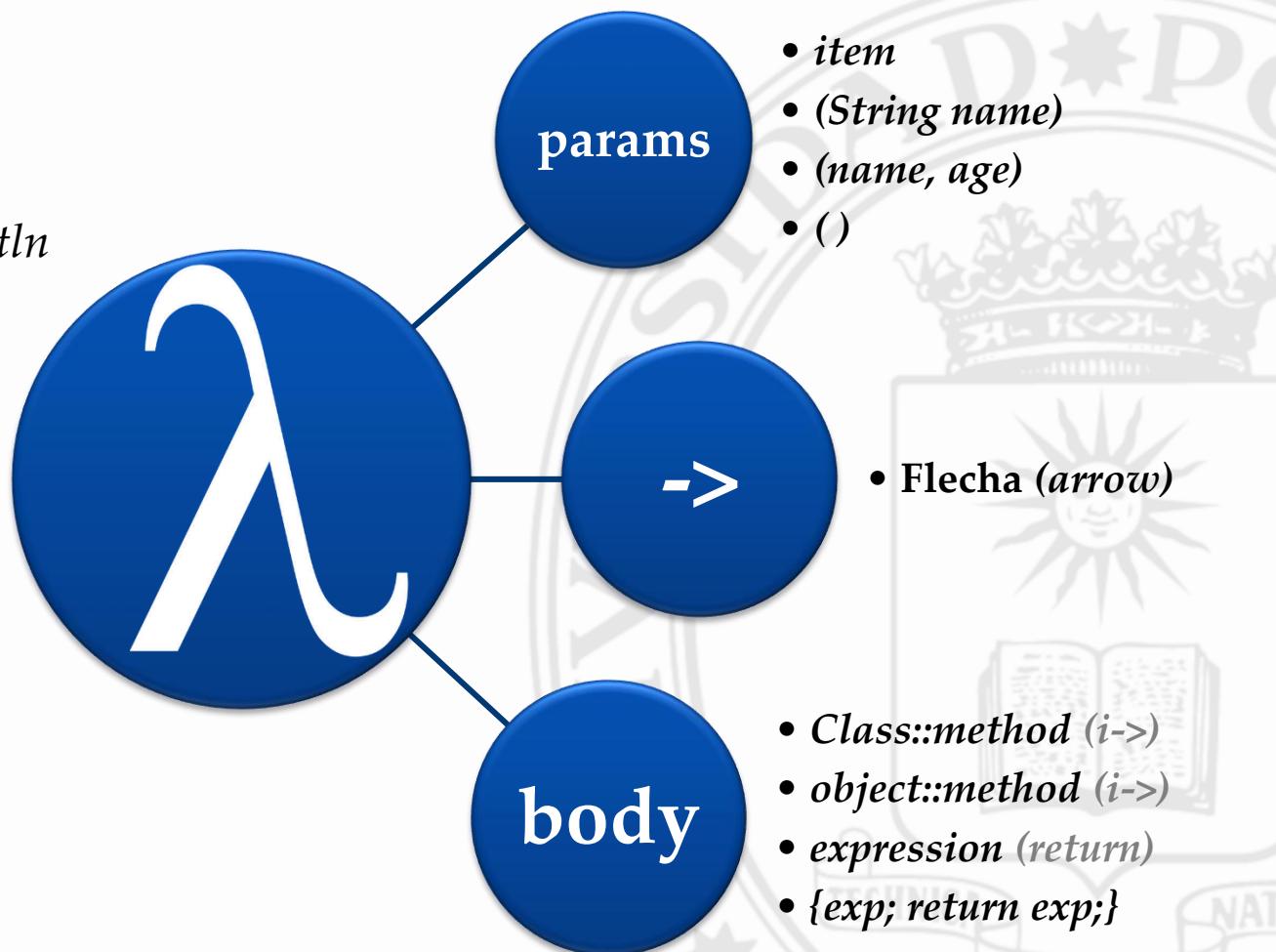
### Servicios (*Model model*)

# Java

## Función Lambda

### Función anónima

- *Alonzo Church*
- *item -> item+1*
- *System.out::println*



# Java

## Función Lambda

Consumer<T> accept(T)

- System.out::println (item->System.out.println(item))

Function<T,R> apply(T):R

- item -> item +1 (item->return item+1)

Predicate<T> test(T):boolean

- item -> item > 0 (item->return item>1)

Supplier<T> get(): T

- ()-> "..."

BiConsumer<T,U,R> apply(T,U):R

- (msg1, msg2) -> System.out.println(msg1 + "," + msg2)

BiFunction<T,U,R> accept(T,U):R

- (x, y) -> x + y

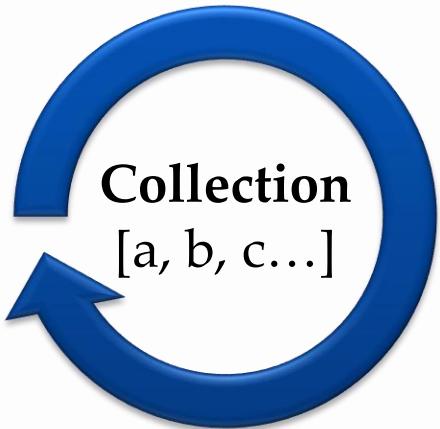
BiPredicate<T,U,R> apply(T,U):R

- String::equals

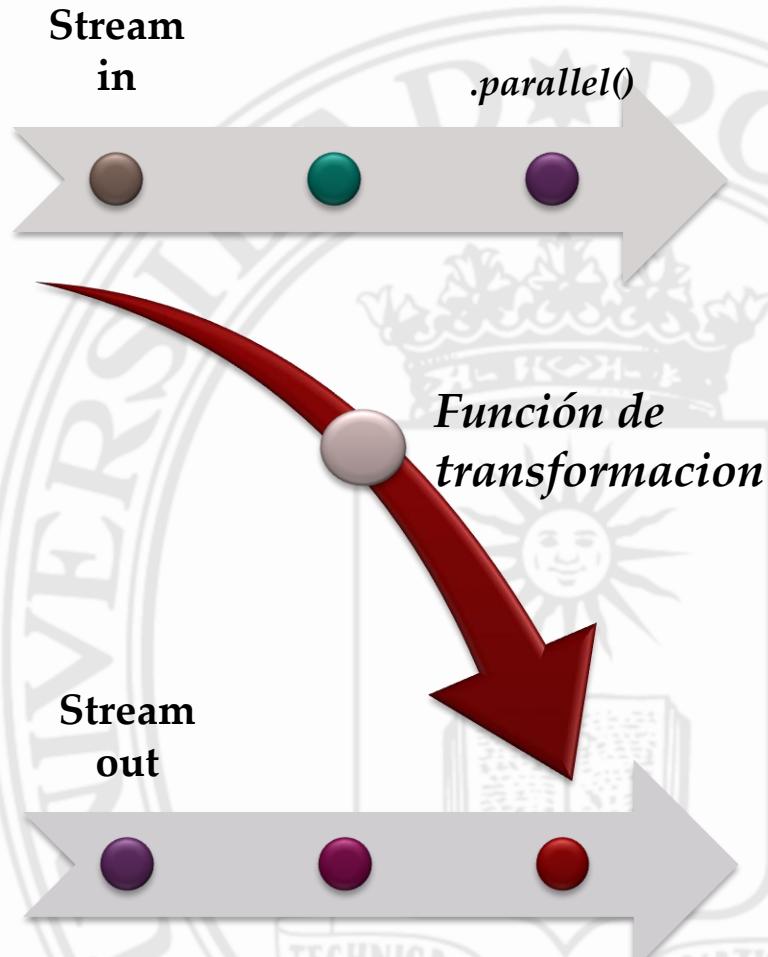
...

# Java

## Collection & Stream



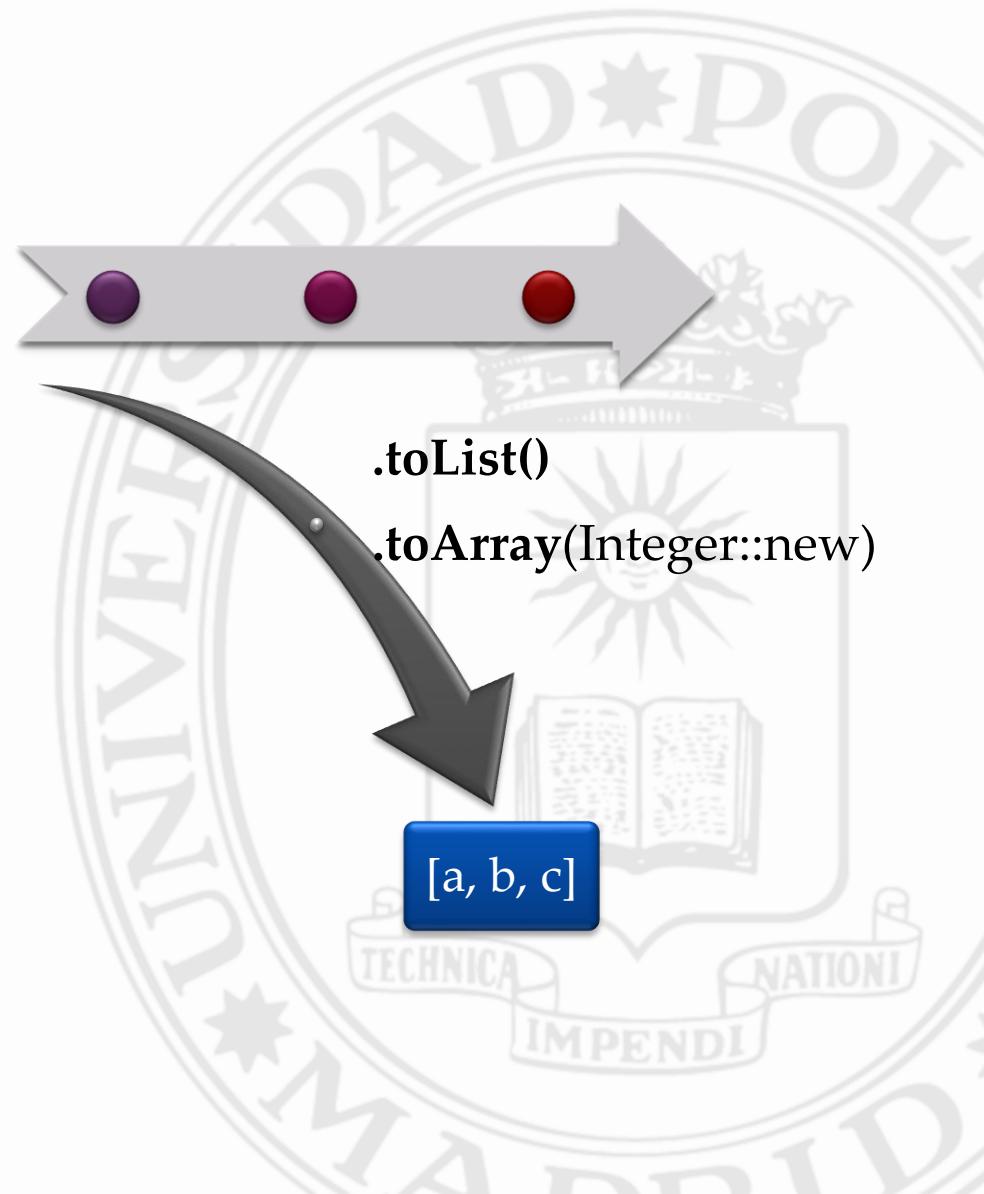
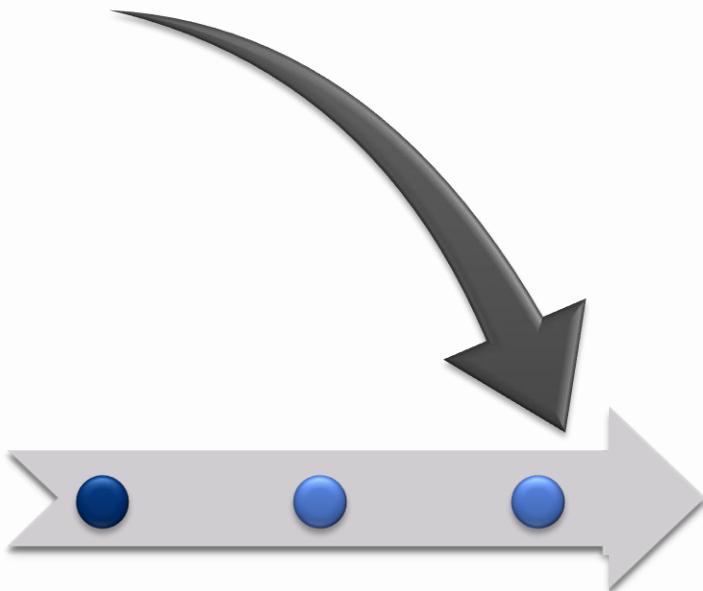
- Gestión bucle
- *Método de transformación*
- Paralelismo
- ¿Reutilización del código?



# Java

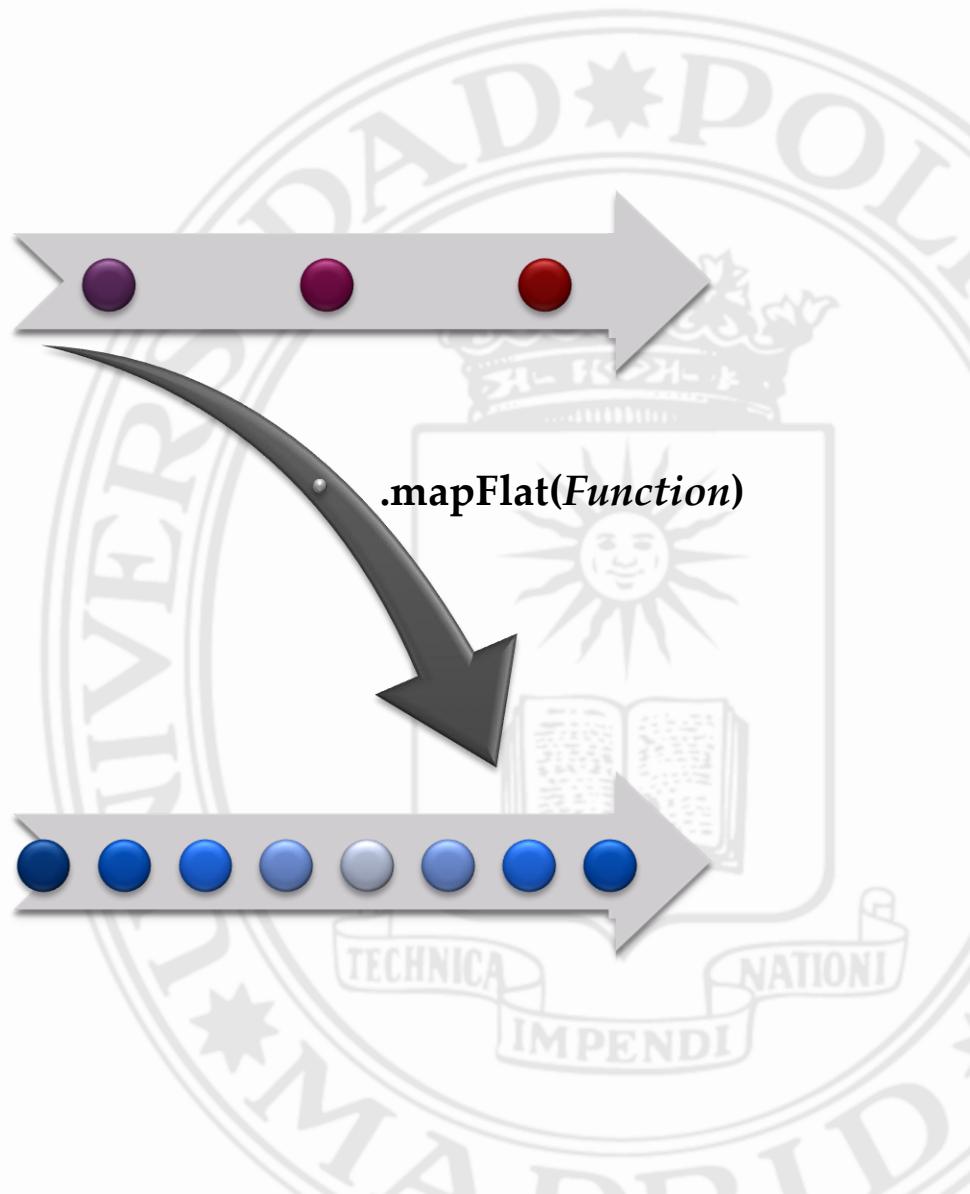
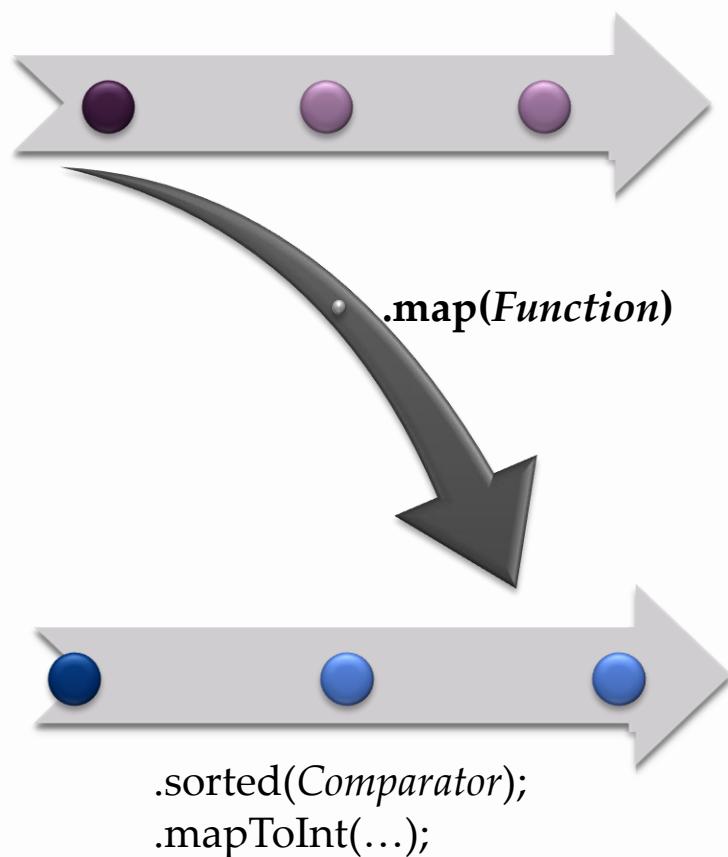
## *Stream. Crear y colección*

```
list.stream();  
IntStream.range();  
Stream.of("1", "2"...);  
Stream.generate();  
Stream.iterate();
```



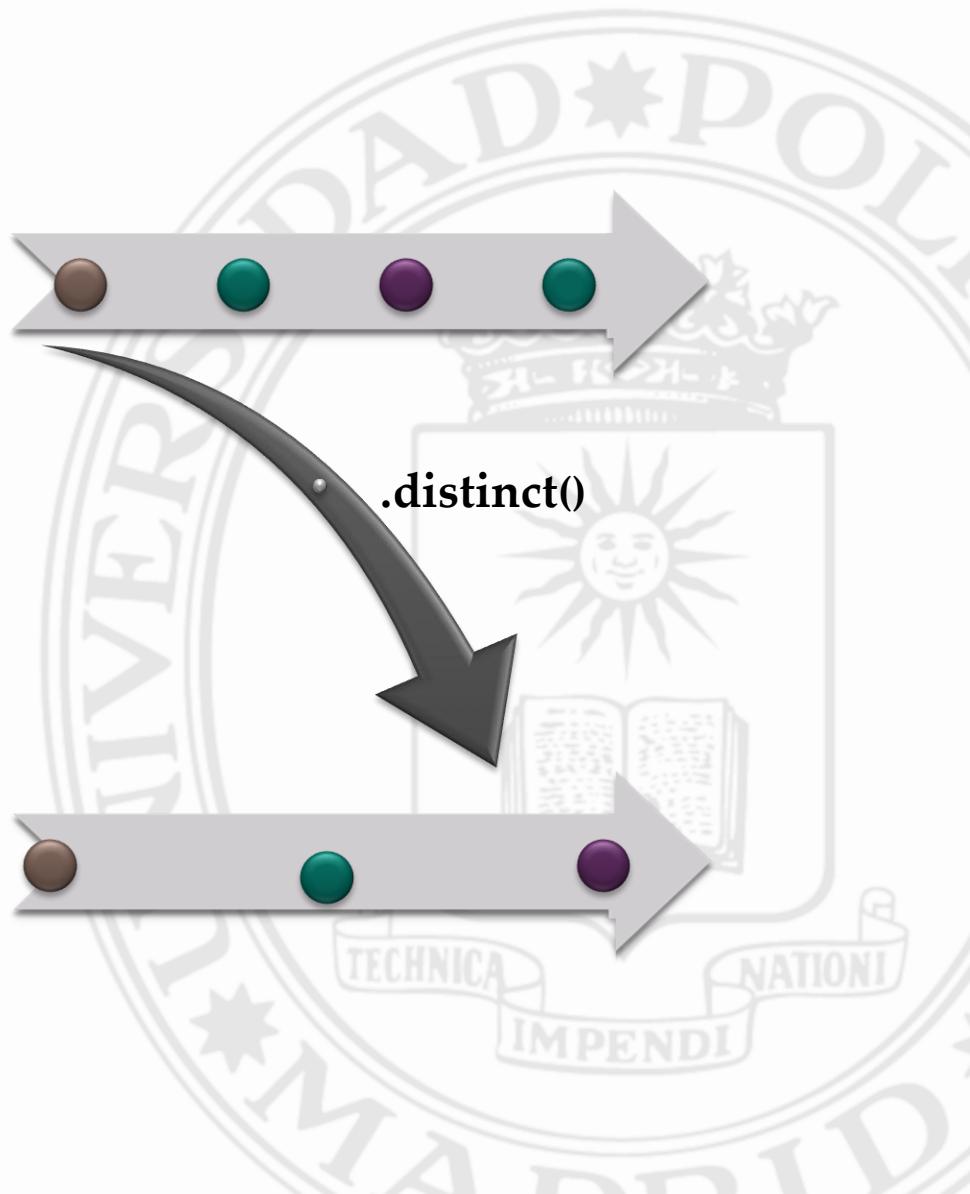
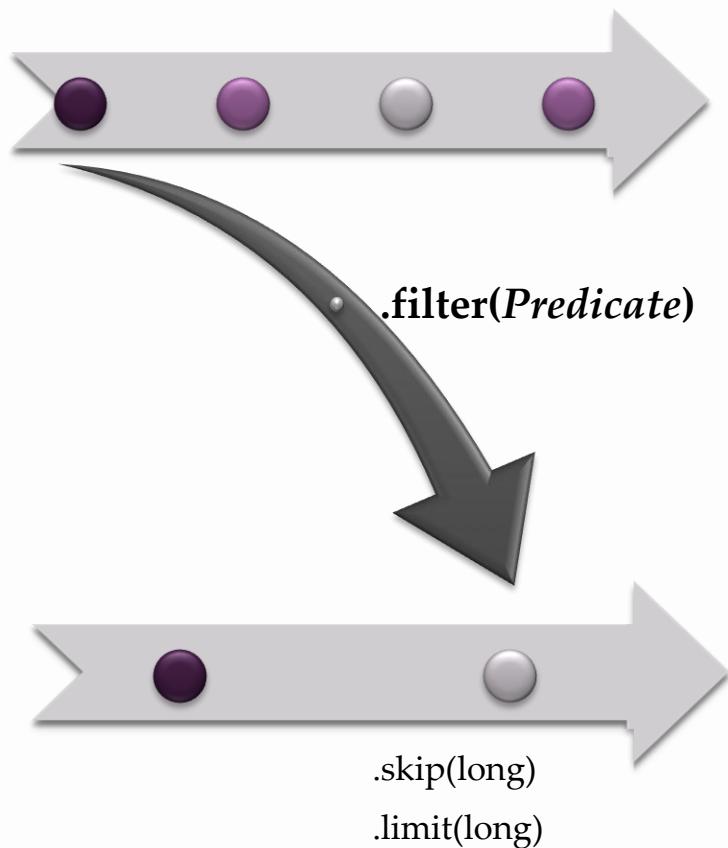
# Java

## *Stream.* Transformación



# Java

## Stream. Filtrado



# Java

## Stream. Operaciones terminales

```
.reduce(Double::sum)  
.reduce(Double::max)  
.reduce(Double::min)  
.reduce(ini,BiFun)  
.forEach():void
```

**.findFirst(*Predicate*)**  
**.anyMatch(*Predicate*)**  
**.noneMatch(*Predicate*)**  
**.allMatch(*Predicate*)**

item

# Code



<https://github.com/miw-upm/iwvg-devops>

- Package: *es.upm.miw.iwvg.devops.code*

## Code in action

- *PointLackOfPOO, Point, DecimalCollection.*
- *Lambda, Flow.*
- *DecimalCollectionWithStream,*  
*DecimalStreamWithFunctionalProgramming.*
- *User, Fraction, UsersDatabase.*
- *Searches.*

## Ejercicios

- *Copiar los ficheros: User, Fraction, UsersDatabase en tu proyecto.*
- *Ampliar Searches.*

# Test Tipos

## Automáticas

- Clases que prueban clases. Cuando parte del código ha sido modificado, se vuelven a lanzar todas las pruebas.

## Cobertura

- % de líneas de código ejecutadas en las pruebas.
- *End-point, servicios: 100%, de media: >80%.*

## Independiente

- El orden de las pruebas es independiente.
- Las pruebas no alteran el sistema. **Cuidado!!!**

## Sencillez

- Prueba una cosa a la vez. **Uso de pobladores**

## Pruebas Unitarias (\*\*Test)

- Se prueba un módulo de código o clase independientemente del resto. Si existen dependencias entre componentes se rompen con los *mocks*.

## Pruebas de Integración (\*\*IT)

- Se prueban los diferentes componentes que dependen de otros componentes. Intervienen varias capas. Pruebas sobre BD.

## Pruebas Funcionales (\*\*FT)

- Se prueba el sistema como un todo. Por ejemplo los *end-point* de un *API Rest*.

## Pruebas de Aceptación

- Los clientes prueban la versión entregada.

# Test JUnit 5

@BeforeAll

- Se ejecuta una sola vez y el método debe ser *static*

@BeforeEach

- Se ejecuta antes de cada uno de los marcados con @Test. Prepara el test

@Test

- Marca un método como prueba

@AfterEach

- Se ejecuta después de cada uno de los @Test. Para liberación de recursos

@AfterAll:

- Se ejecuta al final del proceso completo y el método debe ser static

@Test  
assertJ

```
assertThat(frodo).isIn(sam, frodo, pippin);
assertThat(frodo.getName()).isEqualTo("Frodo").isNotEqualTo("Frodon");
assertThat("The Lord of the Rings").isNotNull().startsWith("The").contains("Lord").endsWith("Rings");
assertThat("Carlos").satisfies(nombre -> {
    assertThat(nombre).hasSize(6).assertThat(nombre).startsWith("C");
});
assertThat(usuarios).extracting(Usuario::getNombre).contains("Carlos", "Ana", "Luis").doesNotContain("Pedro");
```

<https://github.com/miw-upm/iwvg-devops>

- Package: `es.upm.miw.iwvg_devops.code.`

### JUnit in action

- Point: `PointTest`, `DecimalCollection: DecimalCollectionTest`.
- `DecimalCollectionWithStreamTest`,  
`DecimalStreamWithFunctionalProgrammingTest`.
- `FlowTest`, `SearchesTest`.
- Lanzamiento de test con IntelliJ: `todo`, `paquete`, `clase` y `test`.
- Cobertura con IntelliJ.

### Ejercicios

- Crear los test para las clases del paquete: `User & Fraction`.
- Ampliar las funcionalidades de `Fraction`.

# Git

## Sistema de Control de Versiones

### Git

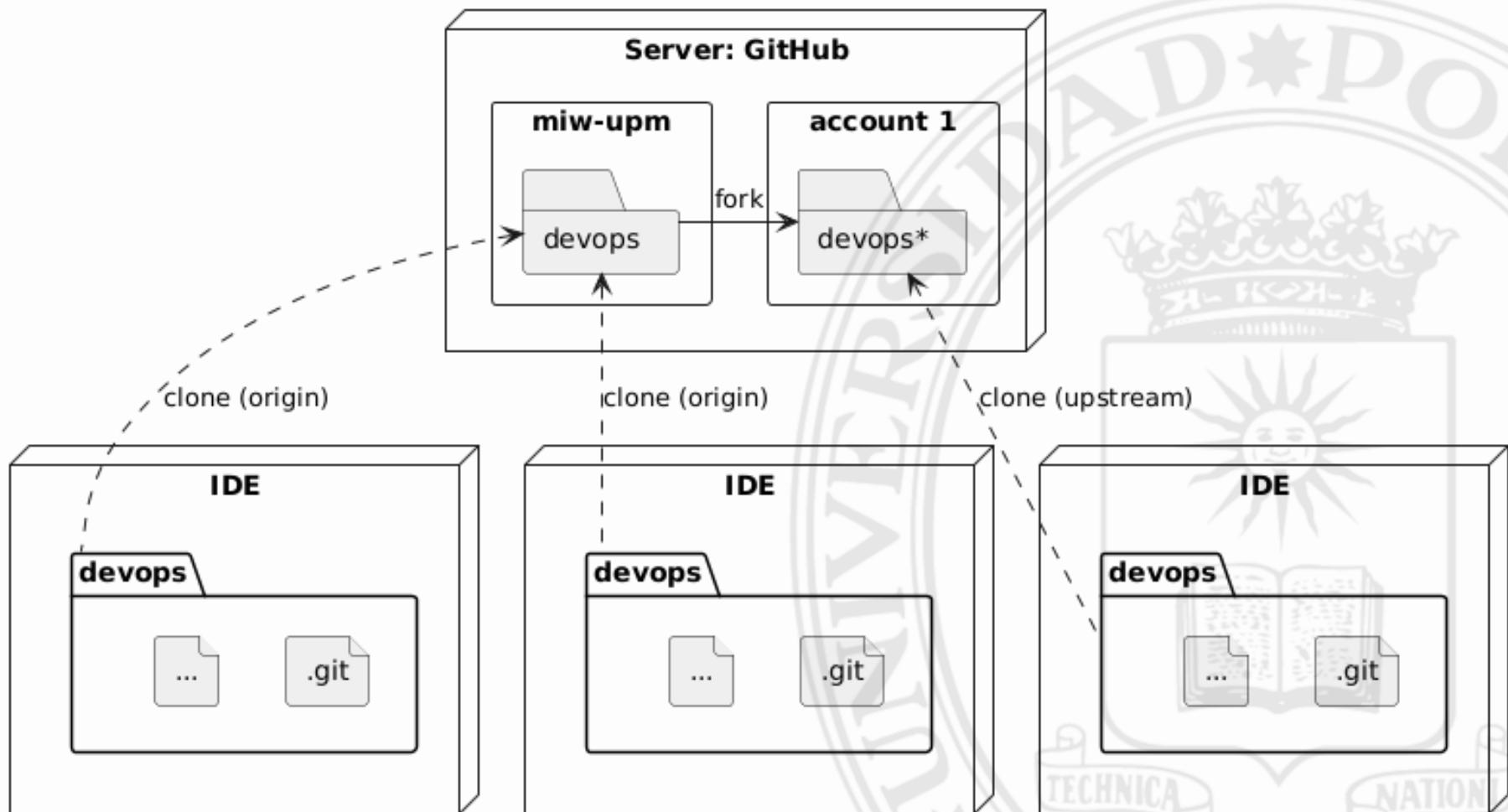
- Nace en 2005.
- Toma como experiencia el proyecto Bitkeeper. (propietario).
- En 2008 nace GitHub, Git Cloud.
- En el 2018 Microsoft adquiere GitHub por 7.500 millones de dólares.
- Software: CLI: <https://git-scm.com>
- *Git Cloud: GitLab, BitBucker...*

### Características

- Control de versiones distribuido.
- Muy fiable, imposible perder el proyecto.
- Trabaja sin necesidad de conexión al remoto, muy rápido. Se podrá sincronizar con el remoto, pero con asistencia...
- **Snapshot:** instantánea (commits)

# Git

## Sistema de control de versiones



# Git IntelliJ

•	updated ci	JBernal	04/03/2025 20:18
•	updated ci	JBernal	04/03/2025 20:09
•	refactoring	JBernal	26/02/2025 13:54
•	changed to configuration in yml	JBernal	26/02/2025 11:00
•	added docker	JBernal	26/02/2025 10:46
•	version 2.1.0-SNAPSHOT	JBernal	26/02/2025 10:18
•	version 1.10.0-Release & Production	origin & master JBernal	26/02/2025 10:09
•	enhancement code	JBernal	26/02/2025 10:06
•	added Functional Test	JBernal	26/02/2025 3:25
•	Spring MVC	JBernal	26/02/2025 3:07
•	refactored	JBernal	21/02/2025 9:24
•	refactored deploy	JBernal	19/02/2025 15:39
•	updated readme	JBernal	19/02/2025 11:07
•	added deployment	JBernal	19/02/2025 11:03
•	version 1.10.0-SNAPSHOT	JBernal	19/02/2025 11:02
•	added deploy on render	release-1.9 JBernal	19/02/2025 10:55
•	version 1.9.0-Release & production	JBernal	19/02/2025 10:42
•	version 1.9.0-Release	JBernal	19/02/2025 10:21
•	update readme	JBernal	19/02/2025 7:47

✓ iwg-devops 2 files

✗ src\main\resources 1 file

application.properties

pom.xml

## version 1.10.0-Release & Production

25525df0 JBernal <j.bernal@upm.es> on 26/02/2025 at 10:09

✓ master, release-1.10 ✗ origin/master, origin/release-1.10

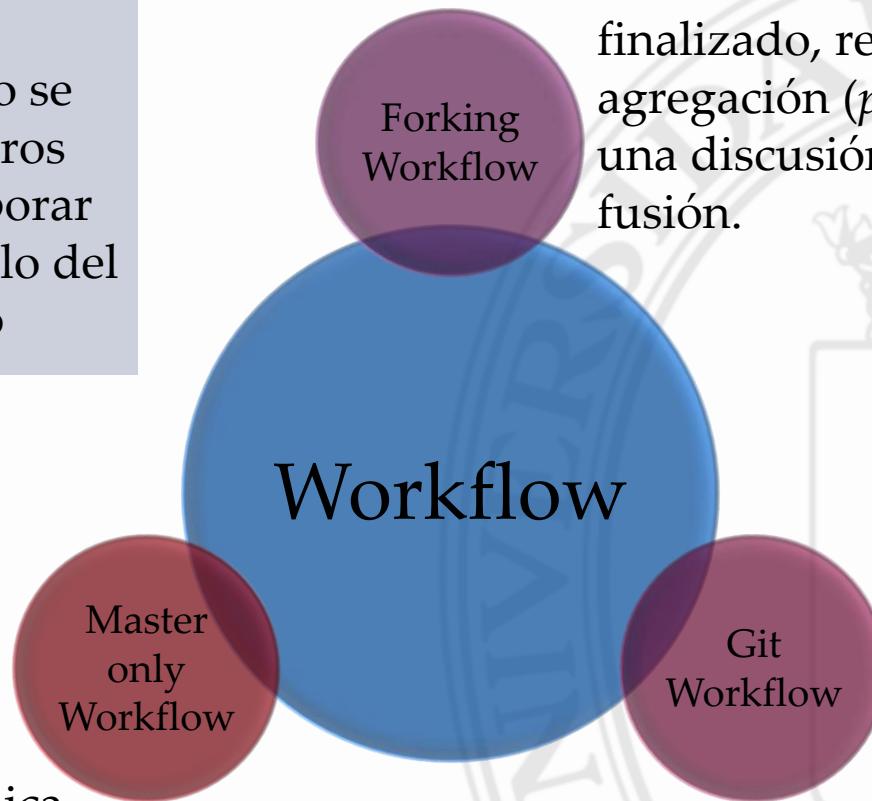
✗ v1.10.0-Release

In 4 branches: master, origin/master, release-1.10, origin/release-1.10

# Git

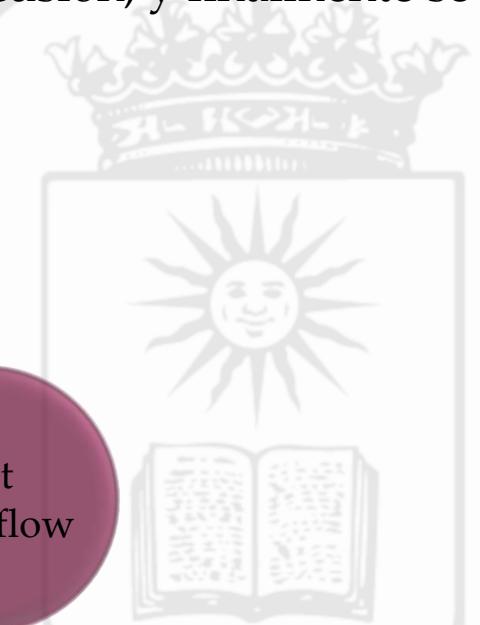
## Flujo de Trabajo

El flujo de trabajo de un sistema de control de versiones indica cómo se relacionan los miembros del equipo para colaborar entre sí en el desarrollo del software colaborativo



Todos comparten la única rama... Fusión problemática!!! y difícil de mantener estable

El usuario bifurca el proyecto y realiza las ampliaciones, una vez finalizado, realiza una petición de agregación (*pull request*). Se abre una discusión, y finalmente se fusión.

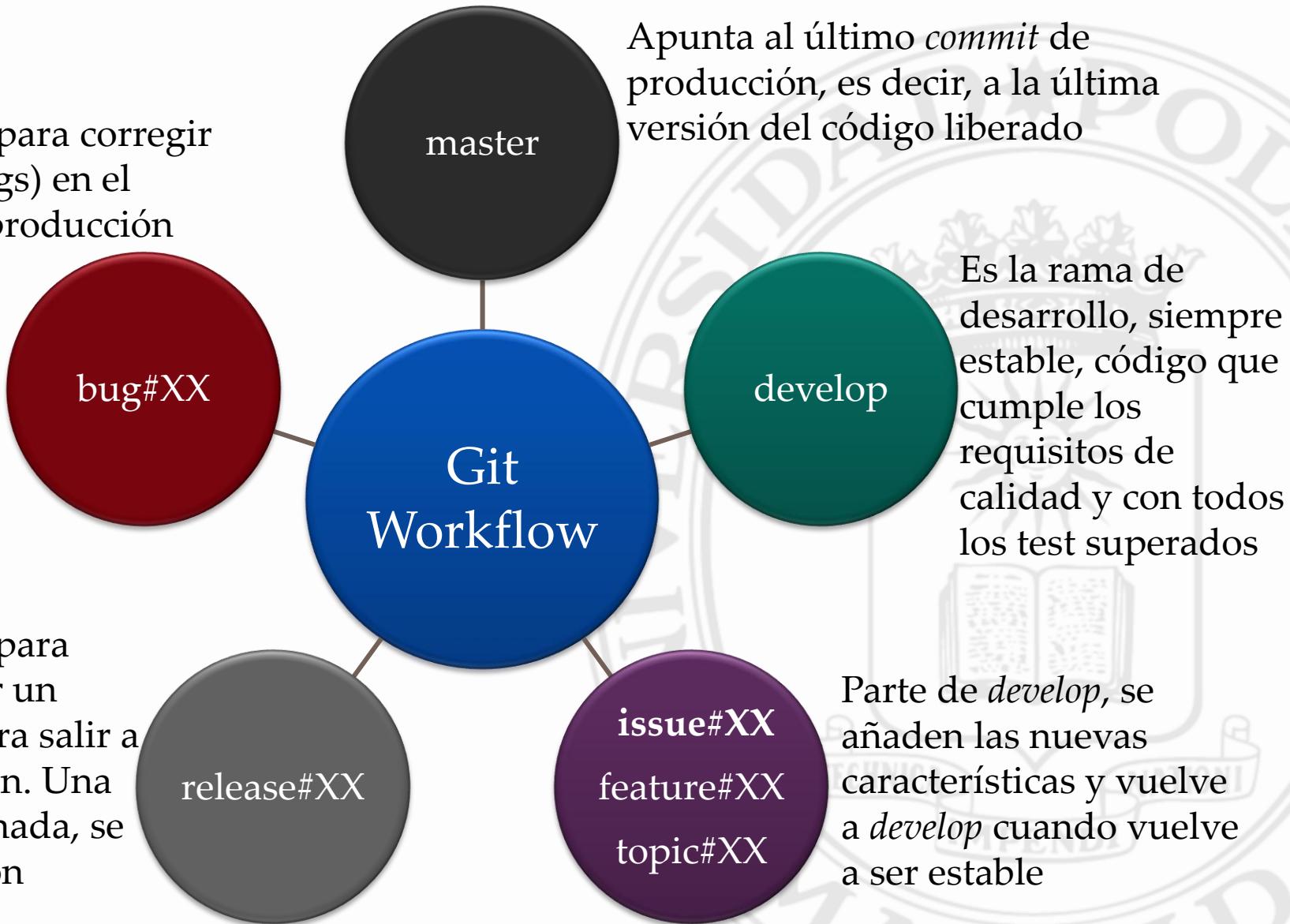


Existen varias ramas con distintas funcionalidades dentro del equipo: *master*, *develop*, *feature*, *release*, *bug*

# Git

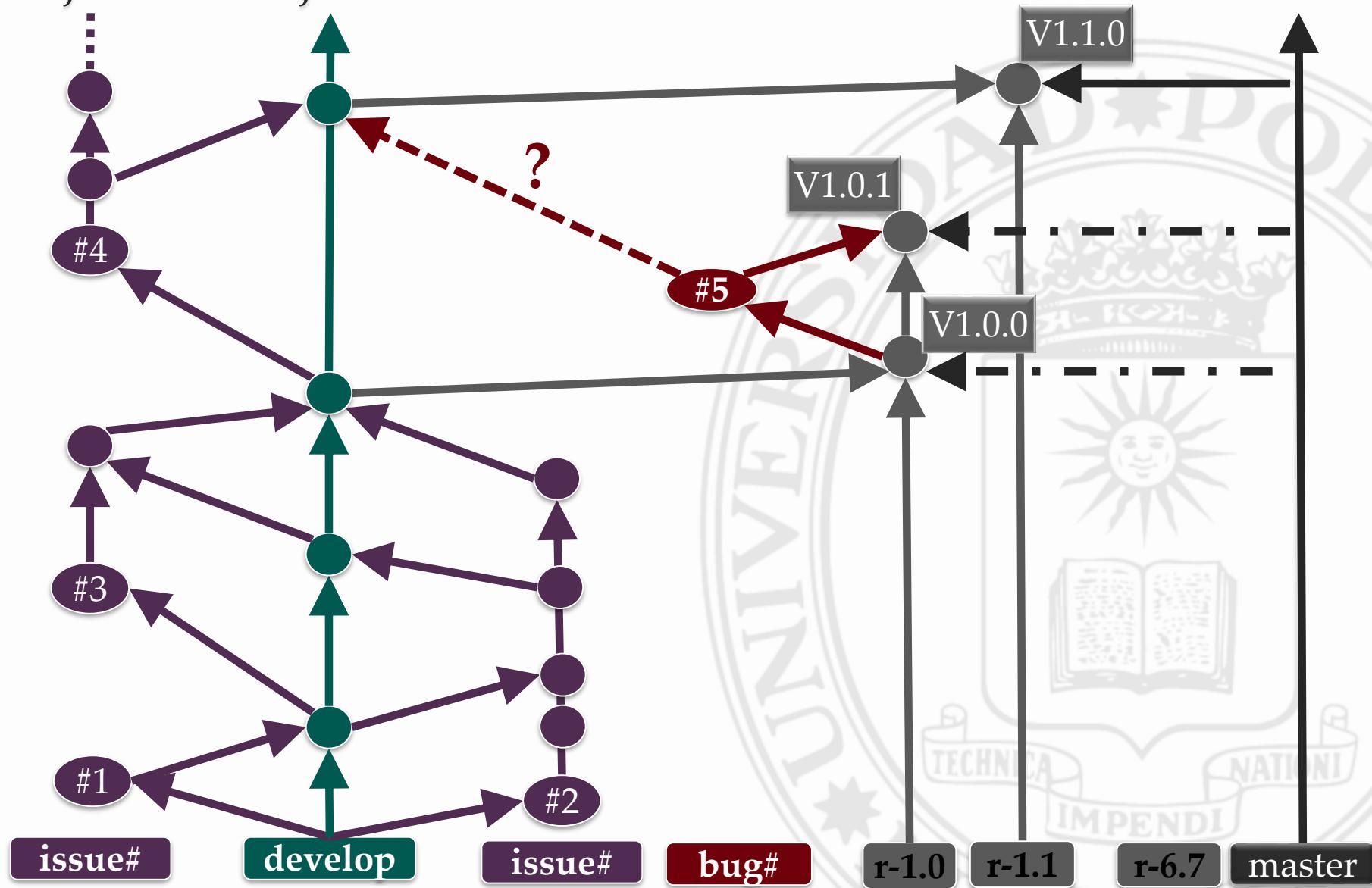
## Flujo de Trabajo Ramificado

Se utilizan para corregir errores (bugs) en el código en producción



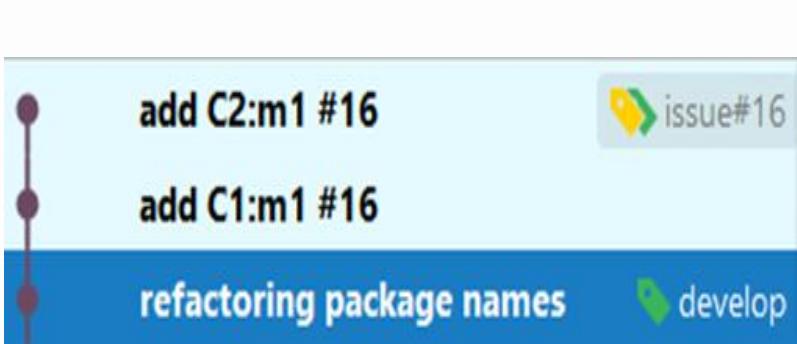
# Git

## Flujo de Trabajo Ramificado



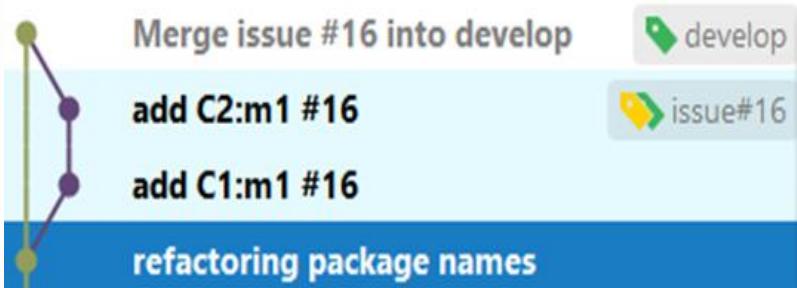
# Git

## Workflow: local



inicio  
issue#

- git checkout -b issue#16
- git add -all
- git commit -m "... #16"
- git commit -m "... #16"



aportación  
parcial

- git checkout develop
- git merge --no-ff -m ".. #16" issue#16
- git checkout issue#16

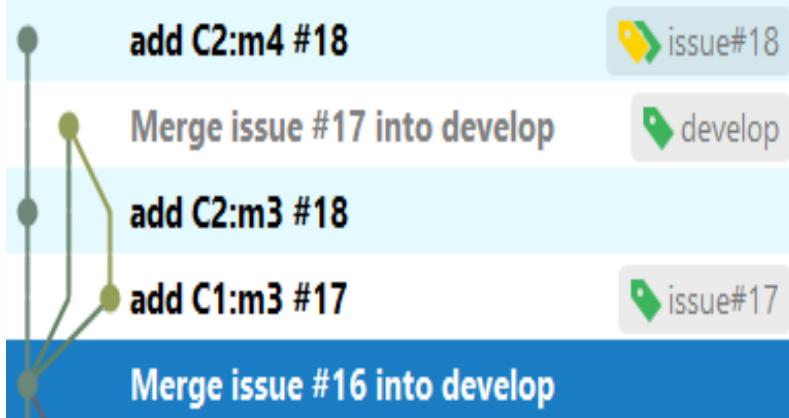


cierre  
issue#

- git commit -m "..."
- git commit -m "..."
- git checkout develop
- git merge --no-ff -m "... #16" issue#16

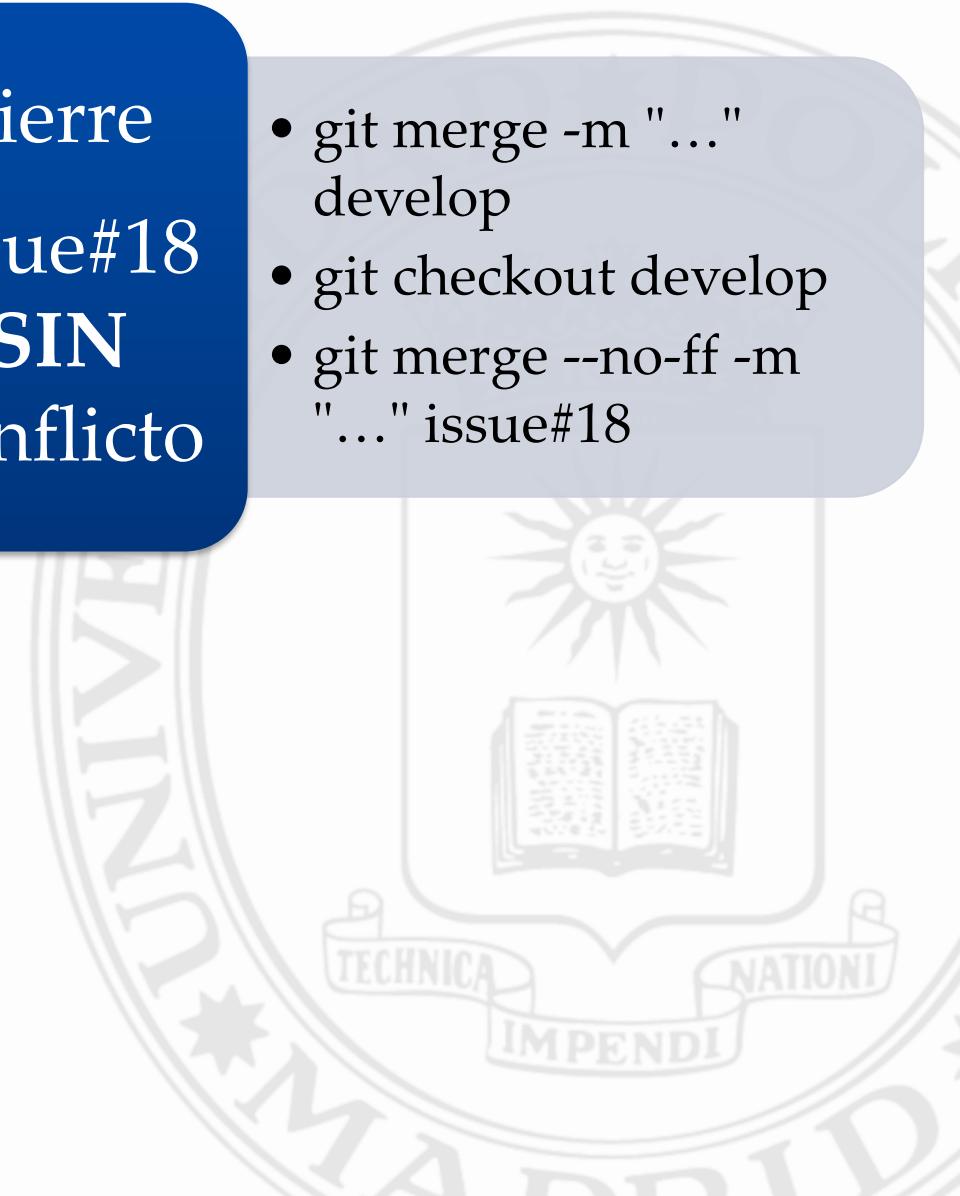
# Git

## Workflow: local



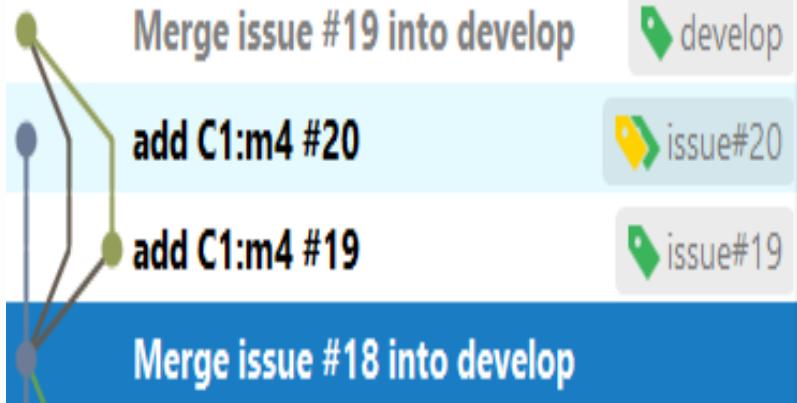
cierre  
issue#18  
**SIN**  
conflicto

- git merge -m "..." develop
- git checkout develop
- git merge --no-ff -m "..." issue#18



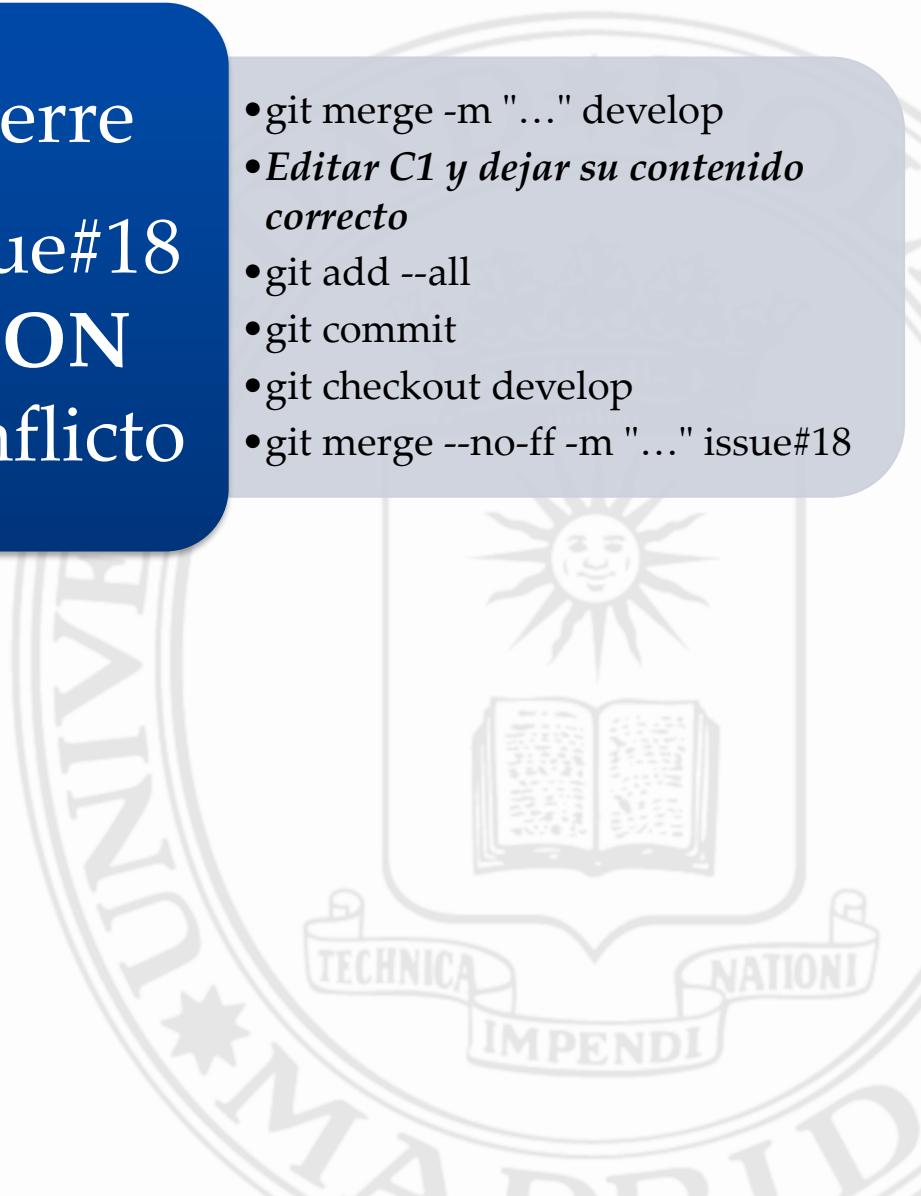
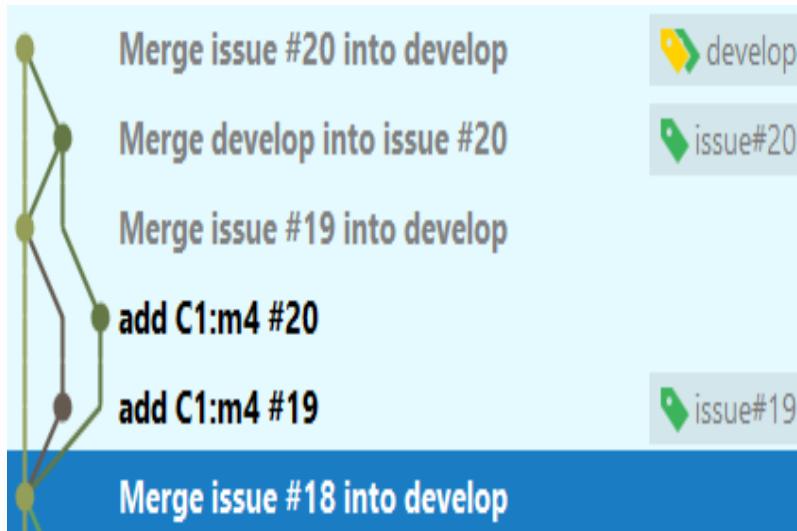
# Git

## Workflow: local



cierre  
issue#18  
CON  
conflicto

- git merge -m "..." develop
- *Editar C1 y dejar su contenido correcto*
- git add --all
- git commit
- git checkout develop
- git merge --no-ff -m "..." issue#18



# Git

## Comandos

### help

- `git help -all`
- `git staus`

### config

- `git config --global user.name "..."`
- `git config --global user.email "..."`
- `git config --list`

### clone

- **git clone <uri>**

### Create

- *git init*
- *git add -all*
- *git commit -m"..."*

### merge

- `git merge --no-ff -m "..." issue#18`

### branch

- `git checkout develop`

# Git



## Clonar el repositorio

- <https://github.com/miw-upm/iwvg-devops>

## Importar desde IntelliJ IDEA

## Git in Action

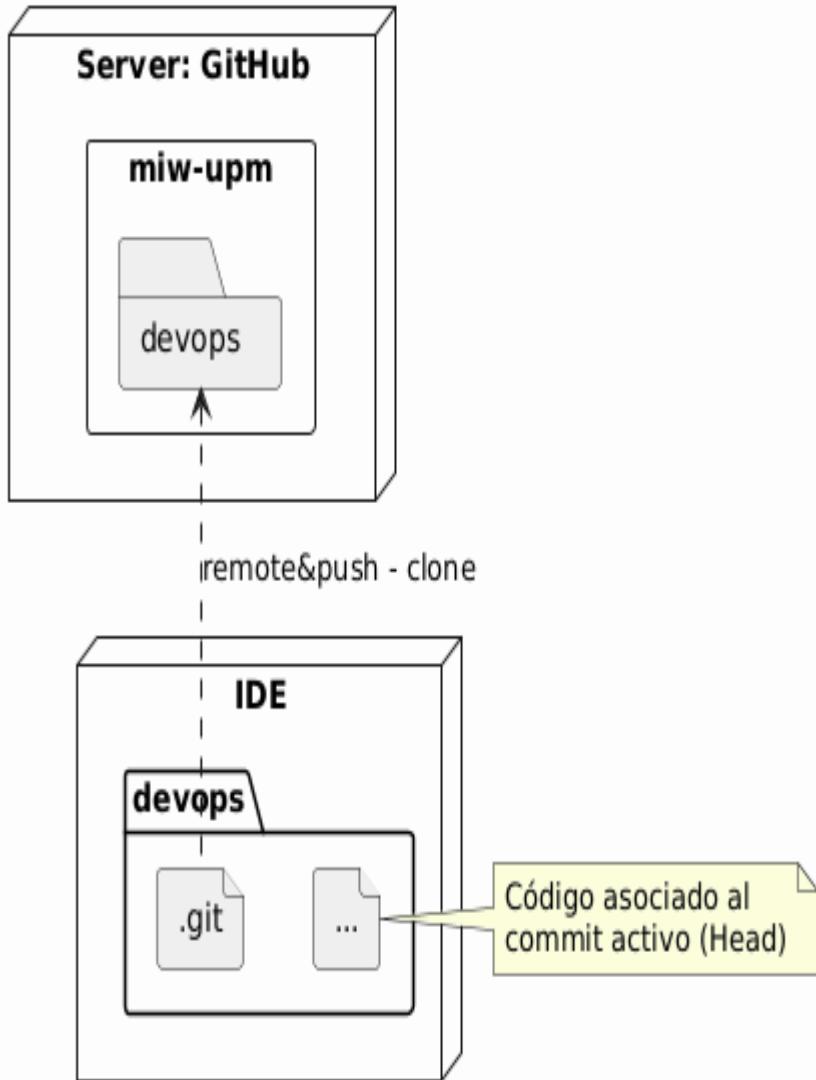
- Crear un proyecto *maven* a partir de la *plantilla* y crear un repositorio local
- Crear la ramas *develop* & *issue#1*
- Trabajar en la rama *issue#1*, creando clases y paquetes
- Incorporar los cambios a la rama *develop*, seguir trabajando en *issue#1* y al final, volver a incorporar a *develop*
- Dos *issues* sin conflictos
- Dos *issues* con conflictos

## Ejercicios

- Crear la rama *issue#2* y repetir lo anterior... repetir con *issue#3...* & *repetir!*
- Crear las ramas *issue#10* & *issue#11*, creando una competencia sin conflicto.
- Crear las ramas *issue#12* & *issue#13*, creando una competencia con conflicto, trabajar en el mismo método.
- Repetir con #14 & #15... *repetir... repetir*

# Git

## Workflow: remoto



### Creación

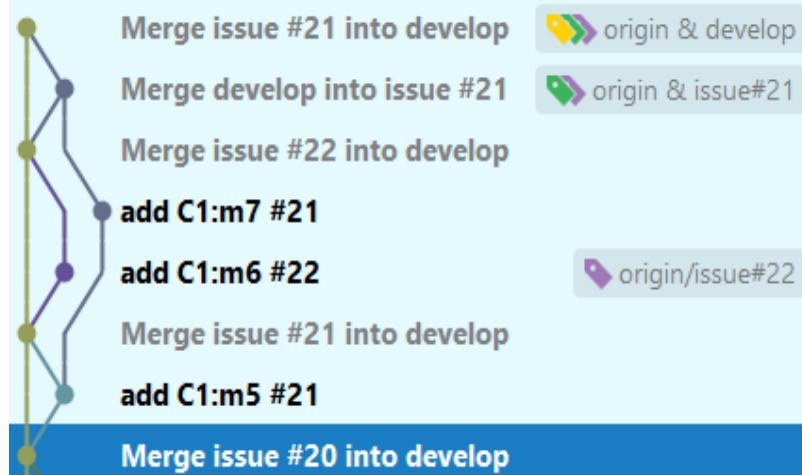
- ...*crear repositorio en GitHub*
- `git remote add origin <URI>`
- `git push origin --all`

### Clonación

- ...*clonar repositorio ya existente*
- `git clone <URI>`

# Git

## Workflow: remoto



cierre  
issue#21

- git fetch origin
- git checkout develop
- git merge origin/develop
- git check issue#21
- git merge -m "..." develop
- git checkout develop
- git merge --no-ff -m "..." issue#21
- git push origin develop

# Git

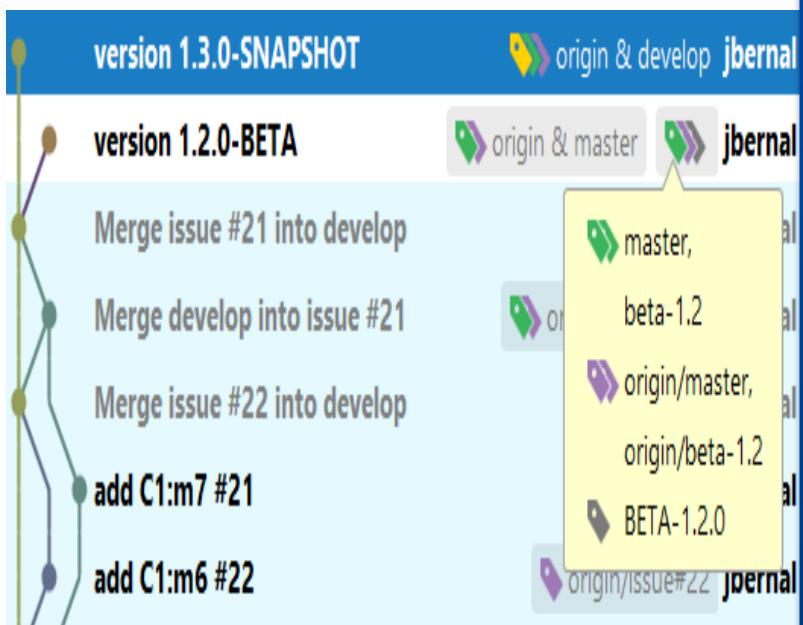
## Workflow: remoto



- Solo FF (**FastForward**)
- git push origin develop

# Git

## Workflow: release

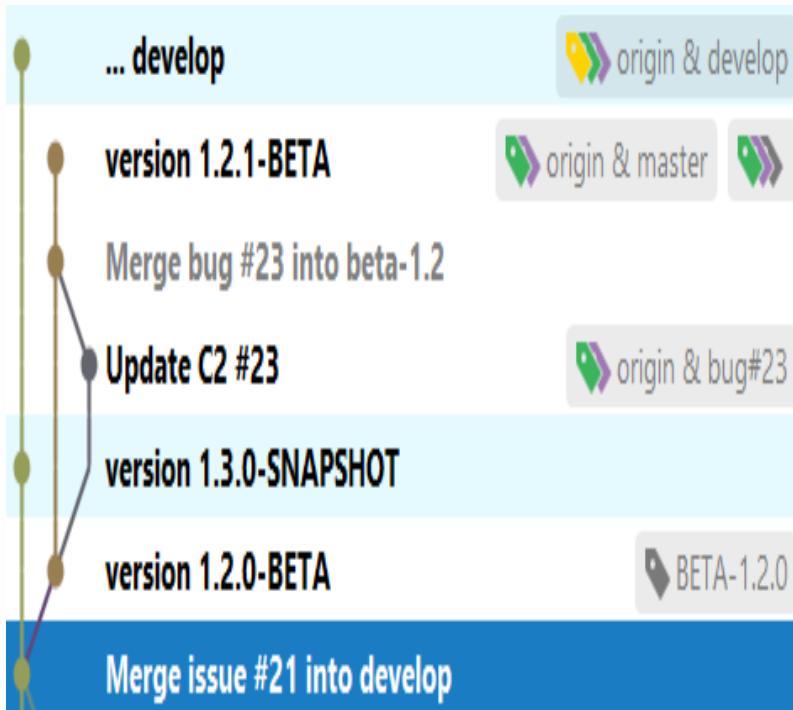


release

- git checkout -b beta-1.2
- ... *change pom.xml*
- git commit -m"..."
- git push origin beta-1.2
  
- git tag -a v1.2.0-beta -m"..."
- git push origin v1.2.0-beta
- git branch -d master
- git branch master
- git push origin master --force
- git checkout develop
- ... *change pom.xml*
- git commit -m"..."
- git push origin develop

# Git

## Workflow: bug



# bug

- git checkout beta-1.2
- git checkout -b bug#23
- ... *corregir problema*
- git commit -m"..."
- git checkout beta-1.2
- git merge --no-ff -m "..." bug#23
- git push origin beta-1.2
- *Liberar nueva versión*

# Git

## Workflow: remoto & Tag

### Consulta

- `git remote -v`
- `git remote rm origin`

### Añadir - Borrar

- **Crear repositorio mediante la Web de GitHub**
- `git remote add origin <url repository>`
- `git remote rm origin`

### Subir - Bajar

- `git push origin <branch>`
- `git fetch origin`

### 💀💀💀 Subir de manera forzada 💀💀💀

- `git push origin <branch> --force`

### Consulta de etiquetas

- **`git tag`**

### Crear etiqueta

- **`git tag -a <etiqueta> -m "mensaje"`**

### Borrado de etiqueta

- **`git tag -d <etiqueta>`**

### Subir etiqueta al remoto

- **`git push origin <etiqueta>`**

# Git

## Workflow: test



Errores?

• Solución?



Error?

• Solución?

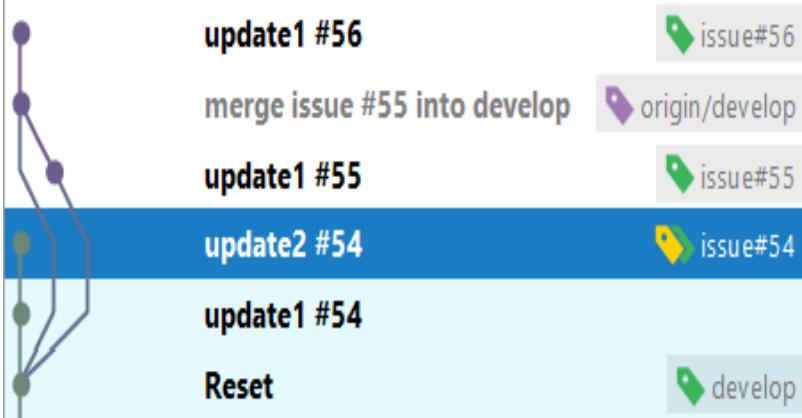


Error?

• Solución?

# Git

## Workflow: test



- Quién #54, #55 o #56?
- Error cometido?
- Solución?

# Git

## Workflow: test

version 2.2.1-Release & jar deploy	origin & master	origin & release-2.2	jbernal	06/09/2021 17:03
Merge #14 into release-2.2			jbernal	06/09/2021 17:01
fixed search 0 #14	origin & bug#14	origin & develop	jbernal	06/09/2021 16:59
version 2.3.0-SNAPSHOT			jbernal	06/09/2021 16:57
version 2.2.0-Release & jar deploy			jbernal	06/09/2021 16:57
Merge #13 into develop			jbernal	06/09/2021 16:55
added search C #13	origin & issue#13	origin & develop	jbernal	06/09/2021 16:55
Merge #12 into develop			jbernal	06/09/2021 16:54
added search 0 #12	origin & issue#12	origin & develop	jbernal	06/09/2021 16:54
Merge #11 into develop			jbernal	06/09/2021 16:54
added search b #11	origin & issue#11	origin & develop	jbernal	06/09/2021 16:53
Merge #10 into develop			jbernal	06/09/2021 16:53
added search 6 #10	origin & issue#10	origin & develop	jbernal	06/09/2021 16:51
version 2.2.0-SNAPSHOT			jbernal	06/09/2021 16:49
version 2.1.0-Release & jar deploy	origin & release-2.1	origin & release-2.1	jbernal	06/09/2021 16:48
Merge #9 into develop			jbernal	06/09/2021 16:46
enhanced FractionTest #9	origin & issue#9	origin & release-2.1	jbernal	06/09/2021 16:46
enhanced Fraction #9			jbernal	06/09/2021 16:45
Merge #6 into develop			jbernal	06/09/2021 16:43
added FractionTest #6	origin & issue#6	origin & release-2.1	jbernal	06/09/2021 16:42
Merge develop into #6			jbernal	06/09/2021 16:41
Merge #8 into develop			jbernal	06/09/2021 13:35
added UsersDatabase #8	origin & issue#8	origin & release-2.1	jbernal	06/09/2021 13:34
Merge #7 into develop			jbernal	06/09/2021 13:33
fixed User #7	origin & issue#7	origin & release-2.1	jbernal	06/09/2021 13:31
added UserTest #7			jbernal	06/09/2021 13:31
added User #7			jbernal	06/09/2021 13:30
added Fraction #6			jbernal	06/09/2021 13:29
Version 2.1.0-SNAPSHOT			jbernal	06/09/2021 13:29
Version 2.0.0-Release & jar deploy	origin & release-2.0	origin & release-2.0	jbernal	06/09/2021 13:27
added Slack #5			jbernal	06/09/2021 13:25
added Heroku #4			jbernal	06/09/2021 13:25
added Better Code Hub #3			jbernal	06/09/2021 13:24
added GitHub badge #1			jbernal	06/09/2021 13:24
fixed Sonarcloud coverage #2			jbernal	06/09/2021 13:23
fixed Sonarcloud #2			jbernal	06/09/2021 13:23
added Sonarcloud #2			jbernal	06/09/2021 13:22
fixed CI error #1			jbernal	06/09/2021 13:22
added CI with GitHub Actions #1			jbernal	06/09/2021 13:21
Initial. Jesus Bernal			jbernal	06/09/2021 12:53

# Git. Workflow



## Git in action

- Liberar la release-1.0.0
- Liberar la release-1.1.0
- Liberar la release-1.0.1
- Liberar la release-1.2.0

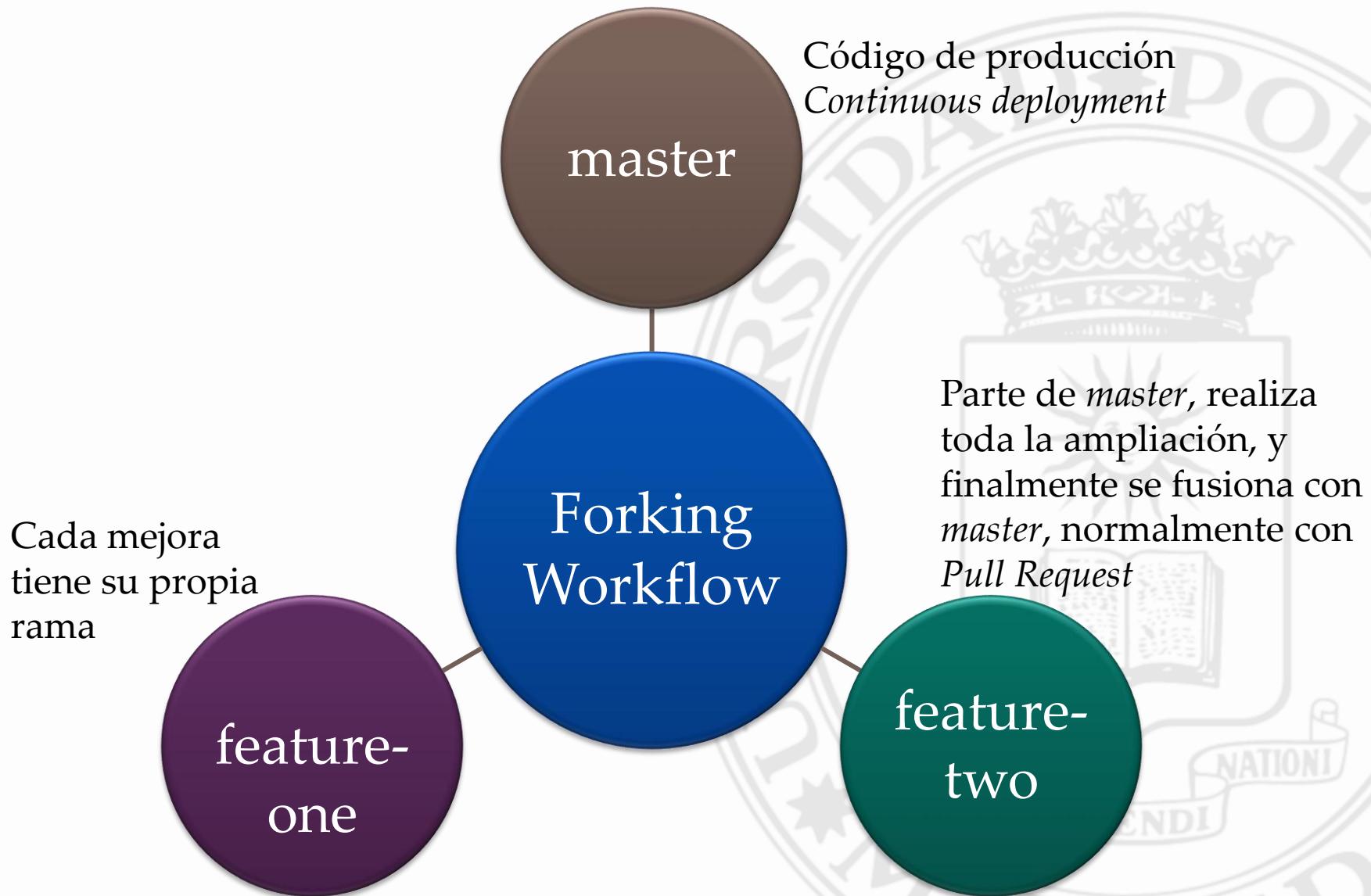


## Ejercicios

- Repetir...

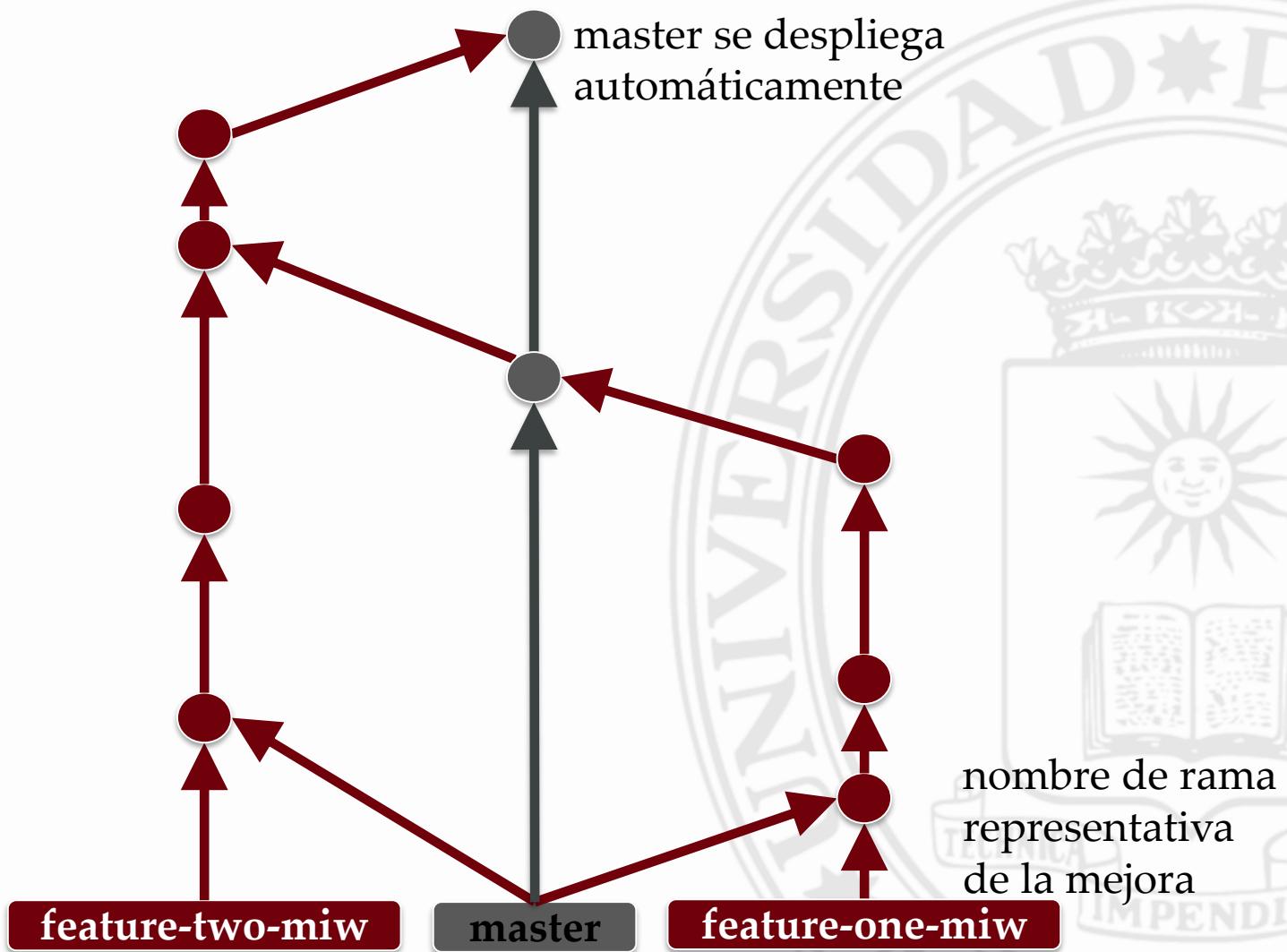
# Git

## Flujo de Trabajo Bifurcación



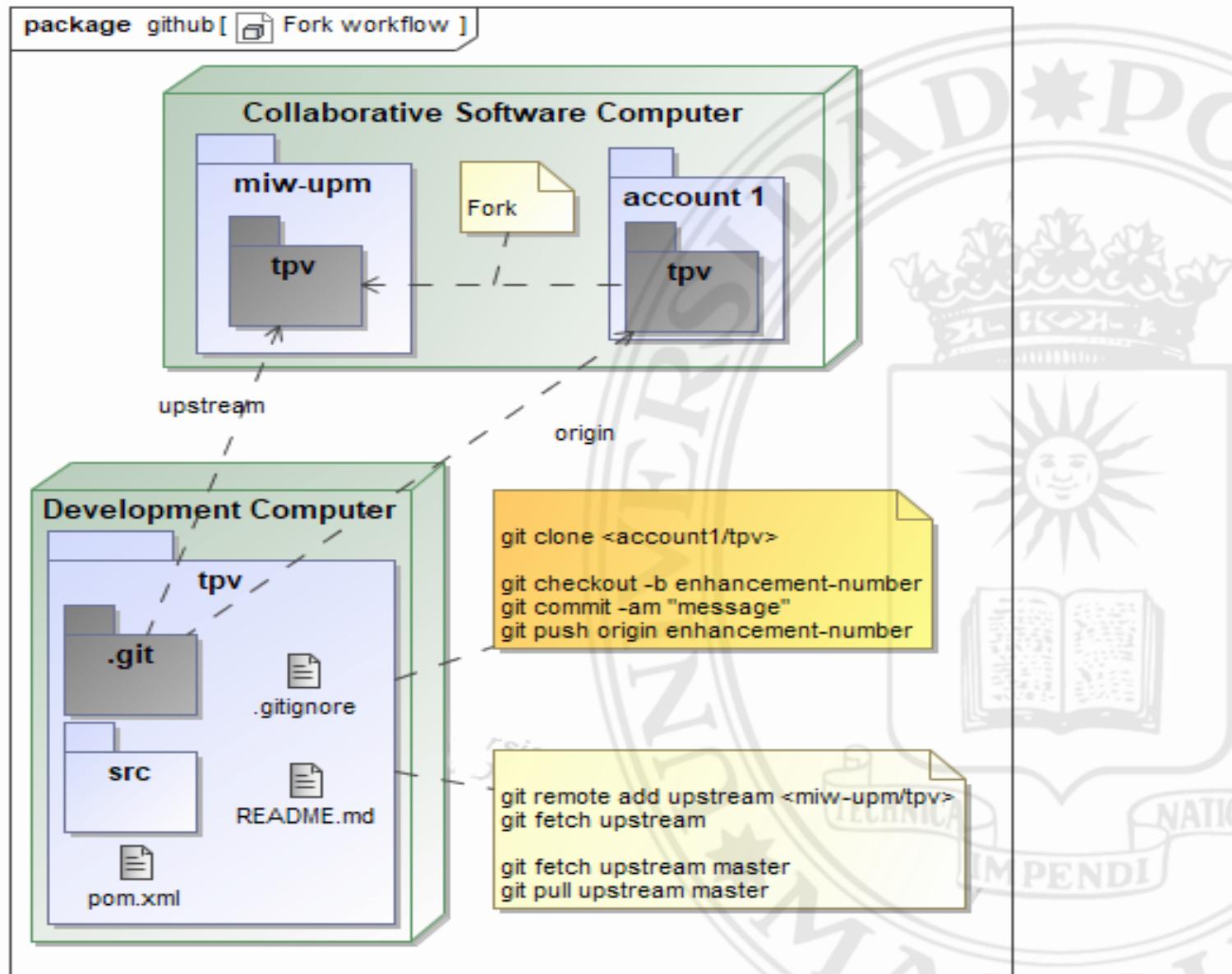
# Git

## Flujo de Trabajo Bifurcación



# GitHub

## Flujo de Trabajo Bifurcación



# GitHub

## Flujo de Trabajo Bifurcación

```
iwvg-fork1 [iwvg-fork-workflow] C:\JBB\work-space
add C2
add C1
add core
Initial
origin & enhancement-number-one
origin & master upstream/master
```

### Forking Workflow

- **git fetch upstream / git pull upstream (con frecuencia)**
- **Nombres de ramas descriptivos**

⚡ jesusBernalBermudez / devops-in-action

forked from miw-upm/devops-in-action

Code

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

⚡ feature-number-one had recent pushes 2 minutes ago

Compare & pull request

master ▾

2 branches

0 tags

Go to file

Add file ▾

Code ▾

This branch is even with miw-upm:master.

⚡ Pull request Compare

# GitHub

## Flujo de Trabajo Bifurcación

# GitHub

## Flujo de Trabajo Bifurcación

### Enhancement number one #1

[Open jesusBernalBermudez... wants to merge 2 commits into miw-upm:master from jesusBernalBermudez:enhancement-number-one](#)

Conversation 1 Commits 2 Checks 0 Files changed 2

jesusBernalBermudez... commented 4 minutes ago

First-time contributor + ...

Enhancement number one Description

miw-upm added some commits 19 minutes ago

- miw-upm add C1 0c4b1d6
- miw-upm add C2 0ac3bb0

miw-upm requested changes just now

View changes

miw-upm left a comment

Owner + ...

Add more methods

Add more commits by pushing to the enhancement-number-one branch on jesusBernalBermudez/iwvg-fork-workflow.

Changes requested 1 review requesting changes [Learn more.](#)

Hide all reviewers

miw-upm requested changes Approve changes Dismiss review Re-request review

Continuous integration has not been set up Several apps are available to automatically catch bugs and enforce style.

This branch has no conflicts with the base branch Merging can be performed automatically.

Merge pull request You can also open this in GitHub Desktop or view command line instructions.

Comment Submit general feedback without explicit approval.

Approve Submit feedback and approve merging these changes.

Request changes Submit feedback that must be addressed before merging.

Reviewers miw miw-upm

Assignees No one—assign yourself

Labels None yet

Projects None yet

Milestone No milestone

Notifications Unsubscribe You're receiving notifications because you commented.

2 participants miw

Lock conversation

Create a merge commit All commits from this branch will be added to the base branch via a merge commit.

Squash and merge The 3 commits from this branch will be combined into one commit in the base branch.

Rebase and merge The 3 commits from this branch will be rebased and added to the base branch.

# GitHub

## Flujo de Trabajo Bifurcación

### Enhancement number one #1

Merged miw-upm merged 3 commits into miw-upm:master from jesusBernalBermudez:enhancement-number-one 10 minutes ago

Conversation 1 Commits 3 Checks 0 Files changed 2 +24 -0

jesusBernalBermu... commented 26 minutes ago Contributor + ...

Enhancement number one Description

miw-upm added some commits 41 minutes ago

MIW add C1 0c4b1d6  
MIW add C2 0ac3bb0

miw-upm requested changes 22 minutes ago View changes

miw-upm left a comment Owner + ...

Add more methods

MIW update C1&C2 b764964

miw-upm merged commit 08096c2 into miw-upm:master 10 minutes ago Revert

Pull request successfully merged and closed Delete branch

You're all set — the jesusBernalBermudez:enhancement-number-o... branch can be safely deleted.

If you wish, you can also delete your fork of miw-upm/iwvg-fork-workflow.

Reviewers MIW miw-upm

Assignees No one assigned

Labels None yet

Projects None yet

Milestone No milestone

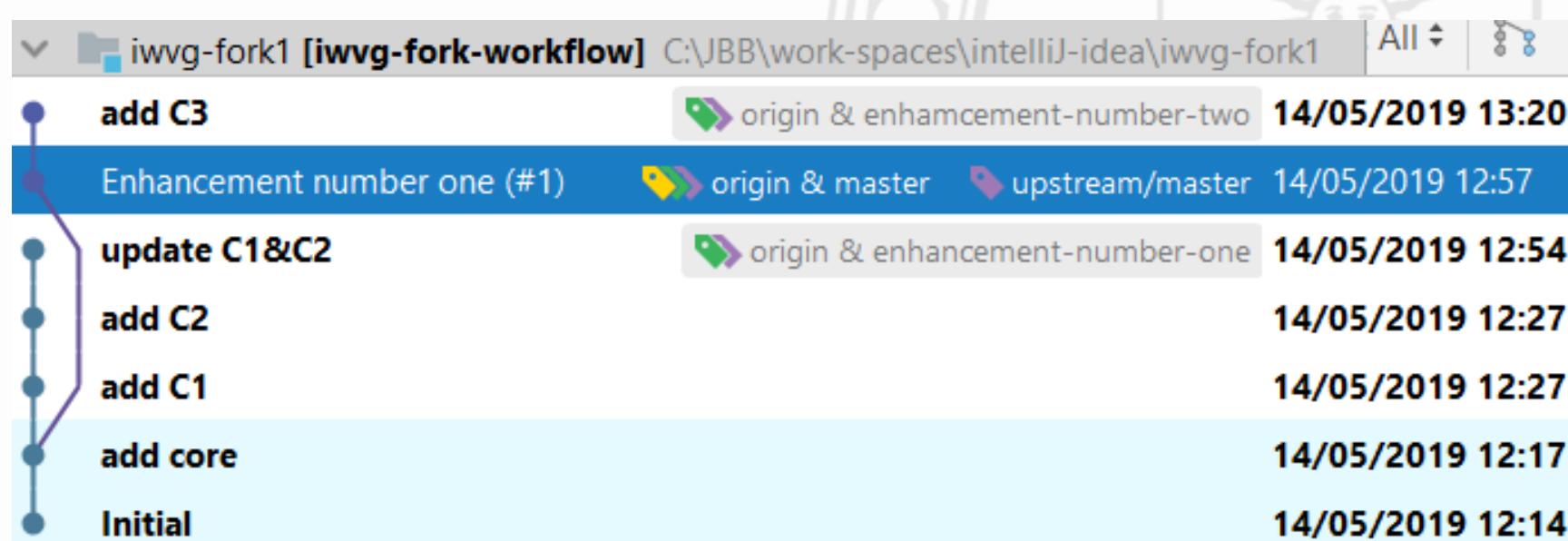
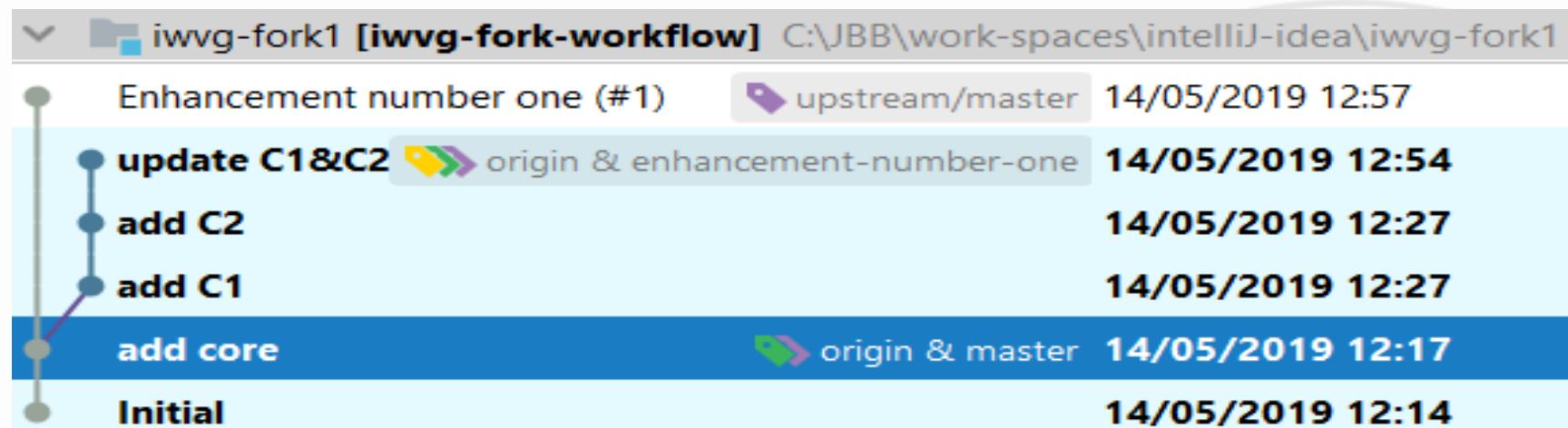
Notifications Unsubscribe

You're receiving notifications because you authored the thread.

2 participants MIW

# GitHub

## Flujo de Trabajo Bifurcación



# Git. Forking Workflow



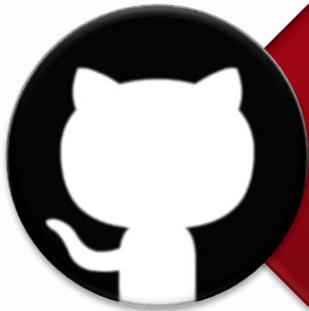
## Git in Action

- *Demo*

## Ejercicios

- Realizar un *fork* al repositorio *miw-upm/devops-in-action*
- Clonar el proyecto en local
  - Asociar remoto *upstream* & *fetch*
  - Crear una mejora en una rama y realizar *pull request*.
  - Actualizar ramas

# Plan



## GitHub



## Creately



## Slack

# GitHub Wiki

The screenshot shows a GitHub Wiki page for the repository 'miw-upm / iwg-devops'. The page title is 'Markdown'. It features a 'New Page' button and a note that 'Máster en Ingeniería Web edited this page 9 minutes ago · 15 revisions'. Below the title, there's a section titled 'IWVG. DevOps. wiki!' with a sub-section 'Lenguaje Markdown'. The page content includes several code snippets illustrating Markdown syntax:

- Línea en blanco para separar estructuras
- \# carácter de escape para los caracteres especiales de Markdown
- # carácter de escape para los caracteres especiales de Markdown

A large blue banner at the bottom of the page contains the word 'Wiki'.

Una Wiki es un sitio web cuyas páginas pueden ser editadas directamente desde el navegador, donde los usuarios crean, modifican o eliminan contenidos compartidos.

GitHub dispone de Wiki para la generación de documentación rápida y compartida a partir de lenguajes de marcado ligeros... *Markdown*.

# GitHub Hitos

The screenshot shows a GitHub repository page for 'miw-upm / iwvg-devops'. The 'Issues' tab is selected, showing 11 issues. Below the tabs are 'Labels' and 'Milestones'. The 'Milestones' tab is active, showing three entries:

- Scrum: sprint 1**: Past due by 11 months, Last updated 12 months ago. Start: 9/01/2019 Team velocity: 50. Progress: 50% complete (2 open, 2 closed). Buttons: Edit, Close, Delete.
- Scrum: sprint 2**: Past due by 11 months, Last updated over 1 year ago. Start: 30/01/2019 Team velocity: 65. Progress: 0% complete (0 open, 0 closed). Buttons: Edit, Close, Delete.
- Curso 2019-20**: Past due by 12 months, Last updated over 1 year ago. Start: MIW-IWVG Software Ecosystem. Progress: 0% complete (0 open, 0 closed). Buttons: Edit, Close, Delete.

At the bottom of the screenshot, the word 'Milestone' is highlighted in blue.

Representa una de fecha de referencia, normalmente asociada a un lanzamiento o finalización de un módulo.

Puede estar abierto-cerrado.

Se les puede asociar tickets.

Tiene marcado un nivel de finalización, obtenido por los tickets asociados que se encuentran cerrados.

# GitHub Issues

The screenshot shows a GitHub Issues page for the repository 'miw-upm / iwg-devops-bernal-jesus'. The page displays 12 issues, with 4 closed and 8 open. The open issues are categorized as follows:

- Bug:** #17 opened on Sep 15, 2022 by miw-upm. Description: Preparación de c...
- Find 4:** #16 opened on Sep 15, 2022 by miw-upm. Description: Preparación de c...
- Find 3:** #15 opened on Sep 15, 2022 by miw-upm. Description: Preparación de c...
- Find 2:** #14 opened on Sep 15, 2022 by miw-upm. Description: Preparación de c...
- Find 1:** #13 opened on Sep 15, 2022 by miw-upm. Description: Preparación de c...
- FractionTest Enhancement:** #12 opened on Sep 15, 2022 by miw-upm. Description: Preparación del s...
- Fraction Enhancement:** #11 opened on Sep 15, 2022 by miw-upm. Description: Preparación del s...

## Issues

Disputa, asunto, tarea, error, ticket...

Se abre un foro de comentarios.

**Importante!!!** en el comentario del *commit* se debe poner una referencia al *issue*:  
“... #17”

# GitHub

## Organización de un Issue

The screenshot shows a GitHub issue page for repository 'Git 11' with issue number #26. The issue is titled 'Ejercicios básicos de GIT. Situaciones anómalas (@miw-upm)' and has a detailed description in the body. The sidebar on the right contains the following information:

- Assignees:** miw-upm
- Labels:** error, type: documentation
- Projects:** None yet
- Milestone:** No milestone
- Notifications:** You're receiving notifications because you're watching this repository.
- Participants:** 1 participant (miw-upm)

At the bottom, there's a text area for comments with 'Write' and 'Preview' buttons, and a 'Leave a comment' button.

### Issue

Breve descripción, en la wiki se detalla. Con la referencia en los *commits* #xx aparece asociado al issue#

# GitHub Scrum

⌘ Ecosystem Software

Backlog | Backlog | By priority | By size | + New View

Filter by keyword or by field

Product Backlog	Sprint Backlog	In progress	Done
iwg-devops-bernal-jesus #13 Find 1 Medium 2 Points User	iwg-devops-bernal-jesus #7 User High 1 Points User	iwg-devops-bernal-jesus #6 Fraction High 1 Points User	iwg-devops-bernal-jesus #3 Análisis del código con Sonarcloud Urgent 0.2 Points Technical
iwg-devops-bernal-jesus #14 Find 2 Medium 2 Points User	iwg-devops-bernal-jesus #8 UserTest High 1 Points User		iwg-devops-bernal-jesus #2 Integración continua con GitHub Actions Urgent 0.5 Points Technical
iwg-devops-bernal-jesus #15 Find 3 Medium 2 Points User	iwg-devops-bernal-jesus #9 UserDatabase High 1 Points User		iwg-devops-bernal-jesus #4 Desplegar en Heroku Urgent 0.5 Points Technical
iwg-devops-bernal-jesus #16 Find 4 Medium 2 Points User	iwg-devops-bernal-jesus #10 FractionTest High 1 Points Technical		iwg-devops-bernal-jesus #5 Liberación de la Release 3.0.0 Urgent 0.5 Points User

+ Add item

Scrum

Aproximación a Scrum: es un proyecto *Automated Kanban*. Los *hitos* representan un *Sprint*. Con la etiquetas de *point*, se establece la estimación ...

## Story

- Dado un caso de uso, crear la historia, cada historia un *issues* en el proyecto. Asociarle *prioridad, estimación, tipo...*

## Backlog

- Mover las historias a la columna *backlog*, son las que desarrollan en el próximo *sprint*, asociarles el *sprint* (hito).

## Sprint

- Cuando alguien inicia una historia (*issue*), se lo asigna y lo mueve a la columna *In progress*. Si es una historia muy grande se podrían plantear varias tareas, cada **tarea** un *issue*.

## Branch: issue#

- Se crea la rama “*issue#xx*” y se programa la historia o tarea... un *commit* debe ser reflejado en el historial del *issue*, se añade en el mensaje “ #xx” o “**miw-upm/repo#xxx**”.

## Fusión

- Cuando el *issue* se termina, se fusiona con *develop*:  
`git merge -no-ff -m "Merge #xx into develop. Detalles" issue#xx`

## Close

- Se cierra la historia-tarea (*issue#xx*), se establece el tiempo consumido y se abandona la rama

## GitHub in action

- Workflow exercises with Scrum

## 📝 Ejercicios

- Crear un proyecto
- Asignaros un par de tickets
- Resolverlos

# Slack

# Comunicación

The screenshot shows a Slack workspace interface. The left sidebar has a dark purple background with various sections: 'miw-upm' (with a dropdown menu), 'Mensajes sin leer', 'Hilos de conversaciones', 'Todos los mensajes directos', 'Menciones y reacciones', 'Slack Connect', 'Más', 'Canales' (expanded), 'iwvg-devops' (selected, highlighted in blue), 'apaw-practice', '# general', '+ Añadir canales', 'Mensajes directos' (with Slackbot, miw-jbernal, Wael Louati, Yosbel Brooks), 'Aplicaciones' (+ Añadir aplicaciones), and 'Canales' (with 'iwvg-devops' expanded). The main area shows the 'iwvg-devops' channel. At the top, there's a search bar with 'Buscar en miw-upm' and a magnifying glass icon. Below it, there's a dropdown menu for 'iwvg-devops' with options like 'Añadir un marcador'. The channel header says 'iwvg-devops' with a lock icon. It displays a message from 'miw-jbernal' at 19:07: 'se ha unido a iwvg-devops.' followed by a message at 19:48: 'añadió una integración a este canal: incoming-webhook'. Below this, a GitHub Actions log for commit 174db42d is shown, detailing the repository 'miw-upm/iwvg-devops', author 'jbernal<j.bernal@upm.es>', and workflow 'DevOps - Tests'. Another message from 'incoming-webhook' at 20:07 says 'Hola desde spring!!'. Further down, a code snippet from 'Lambda.java' is shown, followed by a 'pom.xml' file. At the bottom, a PDF document titled 'miw-iwvg-devops-slides.pdf' (10 MB) is displayed, showing the title 'Ingeniería Web: Visión General -IWVG-' and the logo of the University of Alcalá. The bottom navigation bar includes icons for file operations (New, Open, Save, Print, Copy, Paste, Find, Undo, Redo) and communication (Send, Direct Message, Group Chat).

# Slack

## Instalación



### Slack

- Invitar al espacio de trabajo “*miw-upm*” por invitación



### GitHub → Slack

- Se añade la aplicación de *GitHub* en *Slack*. Se debe conectar con *GitHub* siguiendo los enlaces...
- Escribir en *Slack*: */github help*, para ver los comandos
- En el canal *#xxx*, se activa subscripción escribiendo:
  - */github subscribe miw-upm/iwvg-devops issues,commits,releases*



## Slack in action

- Workflow exercises with Scrum

## 📝 Ejercicios

- Asociar Slack a un proyecto

# ¿Qué es la Integración Continua?

*Continuous Integration*

Práctica del desarrollo del software para integrar código frecuentemente

El objetivo es mejorar la productividad y mantener la calidad del código.

Debe ser un proceso automático tras la detección del cambio del código en el repositorio para encontrar y arreglar los errores con rapidez .

Práctica relacionada con las *metodologías ligeras* y especialmente con *XP*. Evita integraciones infernales. Fue propuesto inicialmente por *Martin Fowler*.

# Principios

Contribuir a menudo: DIA

No contribuir con código roto

Solucionar los test rotos inmediatamente

Mantener la calidad

No continuar con código roto

## Actions



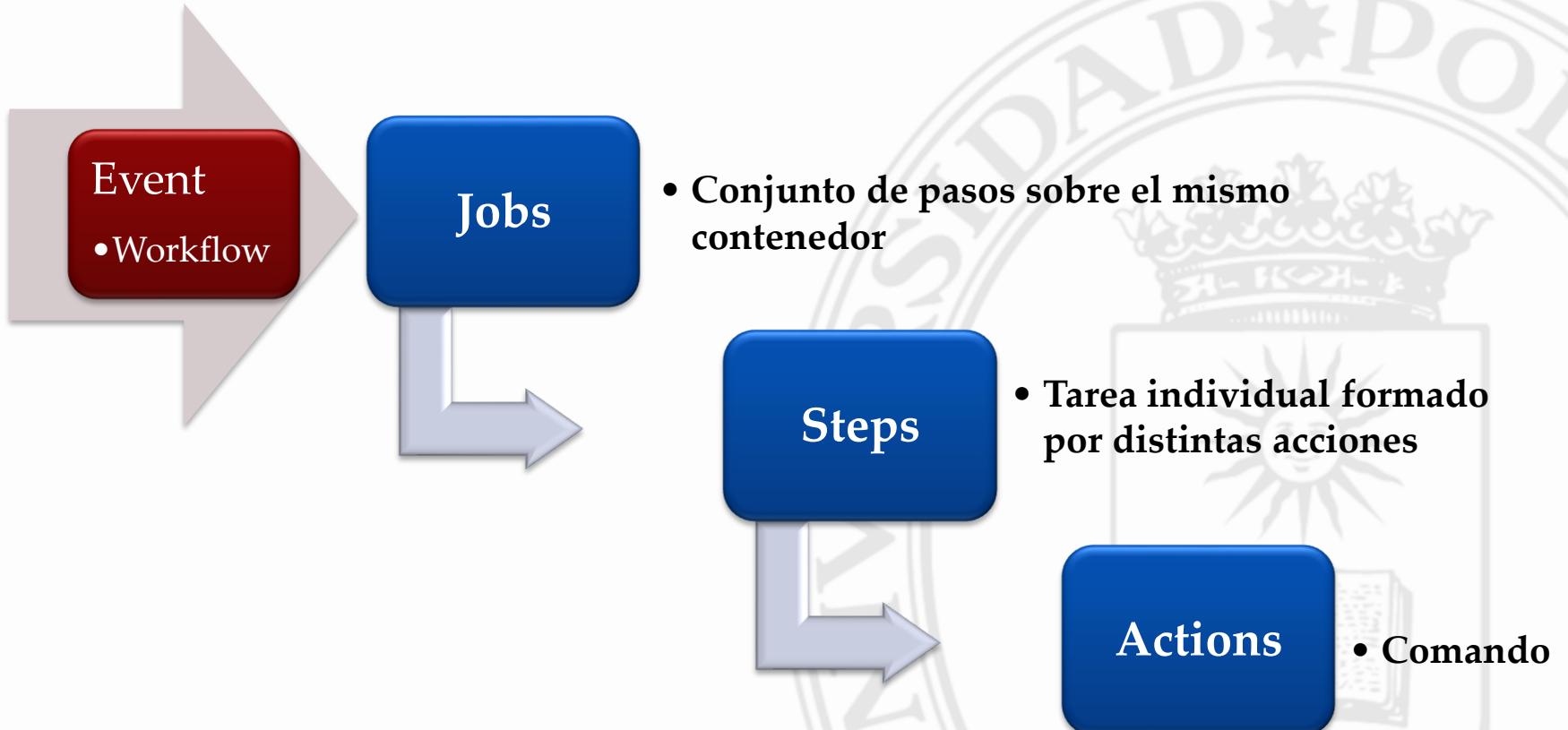
- Servicio de Integración y Despliegue Continuo de GitHub. Basados en flujos de trabajo, con una gran variedad de plantillas y contenedores.
- <https://docs.github.com/en/actions>

## Jenkins



- Jenkins es un software de integración continua de código abierto escrito en Java.

# GitHub Actions



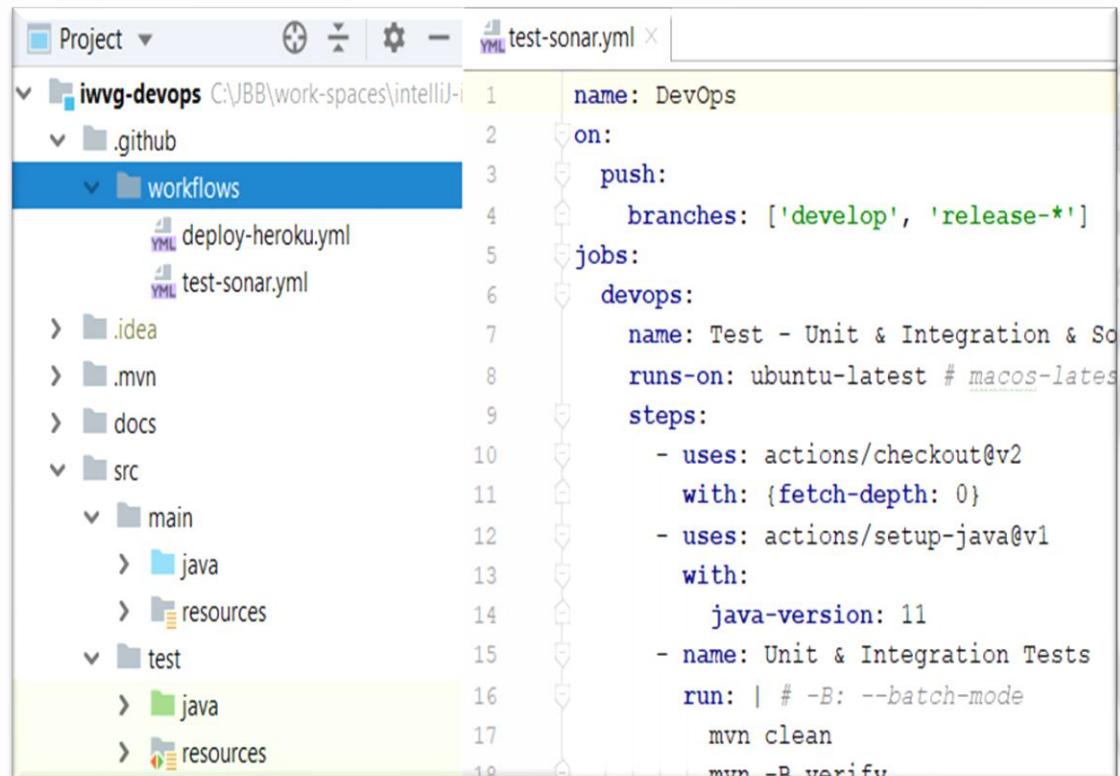
---

Runners    Contenedores (Linux, Windows, macOS)

---

CI

# GitHub Actions



The screenshot shows the IntelliJ IDEA interface with the 'Project' tool window open. The project structure is displayed on the left, showing a folder named 'iwvg-devops' containing '.github', '.idea', '.mvn', 'docs', and 'src' directories. Inside 'src', there are 'main' and 'test' folders, each containing 'java' and 'resources' subfolders. On the right, the code editor displays the 'test-sonar.yml' file, which defines a GitHub Action named 'DevOps' that runs on push events to the 'develop' and 'release-\*' branches. The action involves checking out the repository, setting up Java 11, and running Maven clean and verify commands.

```
name: DevOps
on:
  push:
    branches: ['develop', 'release-*']
jobs:
  devops:
    name: Test - Unit & Integration & SonarQube
    runs-on: ubuntu-latest # macos-latest
    steps:
      - uses: actions/checkout@v2
        with:
          fetch-depth: 0
      - uses: actions/setup-java@v1
        with:
          java-version: 11
      - name: Unit & Integration Tests
        run: | # -B: --batch-mode
          mvn clean
          mvn -B verify
```

## Configuración

Se configura con ficheros **\*.yml**, dentro de la carpeta **.github/workflows**, en la raíz del proyecto.

# GitHub Actions. YAML

## YAML

Key: Value

Structure:  
"whitespace"  
o {"}

List "-" o  
"[\*, \*, \*]"

Comments  
"#"

Single-line  
">" o Multi-  
line "|"

```
test-sonar.yml ×
1   name: DevOps - Tests
2   on:
3     push:
4       branches: [develop, 'release-*']
5   jobs:
6     test:
7       name: Test - Unit & Integration & SonarCloud Scan
8       runs-on: ubuntu-latest # macos-latest macos-11 w.
9       steps:
10      - uses: actions/checkout@v2
11        with: {fetch-depth: 0}
12      - uses: actions/setup-java@v1
13        with:
14          java-version: 11
15      - name: Unit & Integration Tests
16        run: | # -B: --batch-mode
17          mvn clean
18          mvn -B verify
19      - name: Sonar
20        if: success() # always() failure() success()
21        run: |
22          mvn -B verify -DskipTests sonar:sonar
23          -Dsonar.projectKey=es.upm.miw:iwvg-devops
24          -Dsonar.organization=miw-upm-github
25          -Dsonar.host.url=https://sonarcloud.io
26          -Dsonar.login=$SONAR_TOKEN
```

CI

# GitHub Actions

```
name: CI
on:
  push:
    branches: [develop,release-*]

jobs:
  devops:
    name: Tests & CodeQL & SonarCloud
    runs-on: ubuntu-22.04
    permissions:
      contents: read
      actions: read
      security-events: write

    steps:
      - name: ⚡ Checkout repository
        uses: actions/checkout@v4
        with:
          fetch-depth: 0
      - name: 🐾 Set up Java
        uses: actions/setup-java@v4
        with:
          distribution: temurin
          java-version: 21
          cache: Maven
      - name: ⚡ Extract artifact from pom
        run: echo "ARTIFACT=$(mvn help:evaluate -Dexpression=project.artifactId -q -DforceStdout)" >> $GITHUB_ENV
      - name: ✅ Run Unit & Integration Test
        run: mvn -B verify
```

CI

# GitHub Actions

miw-upm / iwg-devops

Unwatch 3 Star 14 Fork 55

Code Issues 14 Pull requests Discussions Actions Projects 1 Wiki Security ...

Workflows New workflow

All workflows

Showing runs from all workflows

Filter workflow runs

6 workflow runs	Event	Status	Branch	Actor
updated yaml DevOps - Tests #17: Commit aad3881 pushed by miw-upm	develop	3 minutes ago	2m 11s	...
removed code smell DevOps - Tests #9: Commit 9adf44a pushed by miw-upm	develop	17 hours ago	2m 16s	...
removed test error to slack DevOps - Tests #8: Commit 2ee48a9 pushed by miw-upm	develop	17 hours ago	2m 14s	...
added test error to slack DevOps - Tests #7: Commit 174db42 pushed by miw-upm	develop	17 hours ago	41s	...
removed code smell DevOps - Tests #6: Commit f5ee8c9 pushed by miw-upm	develop	18 hours ago	2m 14s	...
CI with github actions DevOps - Tests #1: Commit 18dcd16 pushed by miw-upm	develop	19 hours ago	1m 51s	...

CI

# GitHub Actions

miw-upm / iwg-devops

<> Code Issues 14 Pull requests Discussions Actions ...

**updated yaml** ...

✓ DevOps - Tests #17

**Test - Unit & Integration & SonarCloud Scan & Slack WebHook** ▾ ⚙ succeeded 3 minutes ago in 1m 58s

Step	Description	Time
>	Set up job	3s
>	Run actions/checkout@v2	3s
>	Run actions/setup-java@v1	5s
>	Unit & Integration Tests	46s
>	Sonar	1m 0s
>	Slack WebHook	0s
>	Post Run actions/setup-java@v1	0s
>	Post Run actions/checkout@v2	1s
>	Complete job	0s

# GitHub Actions - CI



## GitHub Actions - CI in action

- Integración continua

## 💡 Ejercicios

- Integrar GitHub Actions-CI con un proyecto

## GitHub Actions → Slack



- Se necesita una aplicación para conectar: *WebHook*
- Se elige el canal, y en la configuración se añade la App: *WebHook*: se debe conseguir la URL
- Se debe establecer la variable de entorno: `SLACK_WEBHOOK_URL` en el proyecto de GitHub, *Settings/secret*, y el *valor* es la *URL*

# Slack Integración

```
- name: 📲 Notify Slack on failure
  if: failure()
  uses: 8398a7/action-slack@v3
  with:
    status: ${{ job.status }}
    fields: repo,author,commit,author,workflow,
  env:
    SLACK_WEBHOOK_URL: ${{ secrets.SLACK_WEBHOOK_URL }}
```

incoming-webhook APP 19:53

Failed GitHub Actions

8398a7@action-slack

repo	commit
miw-upm/iwvg-devops	174db42d
author	workflow
jbernal<j.bernal@upm.es>	DevOps - Tests

# ¿Qué es el Análisis Estático?

## Análisis Estático

Es un análisis del código sin ejecución.

Nos ayudan a detectar problemas del código fuente, buscando errores de seguridad, código duplicado, problemas potenciales por mala práctica...

# Análisis Estático



## *Sonarcloud*

- Se ofrece en la nube y se integra *GitHub*

# Análisis Estático

## Sonarcloud

Crear cuenta en Sonarcloud

Crear una nueva organización

- Debéis crear una organización, y para ello es mas fácil en la pestaña **Import from GitHub**, crearla mediante el botón: **Choose one of your GitHub**.

Crear un proyecto

- + **Analyze new Project** (menú de la parte superior derecha)
- **créate a Project manually**, copiar Project key: **miw-upm\_iwvg-devops**
- **Set up project for Clean as You Code**: Number of days: 1

Configurar la rama por defecto en Sonarcloud

- Entrar en el proyecto, y pulsar Branches, Renombrar develop si esta no es la rama por defecto
- Renombrar las long living branch: (master | develop | release-.\* )

Generar ApiKey

- [https://sonarcloud.io/account/security/.](https://sonarcloud.io/account/security/)

Guardar ApiKey en GitHub

- **Crear variable de entorno en GitHub en el repositorio, mediante Settings/Secrets, llamada: SONAR\_TOKEN.**

Integrar con Actions

# Análisis Estático

## Sonarcloud. GitHub actions

- **name:** ⚒ Extract artifact from pom

**run:** echo "ARTIFACT=\$(mvn help:evaluate -Dexpression=project.artifactId -q -DforceStdout)"  
-> \$GITHUB\_ENV

- **name:** 📊 Sonar Analysis

**run:** >

mvn -B sonar:sonar

-Dsonar.projectKey=miw-upm-github\_\${{ env.ARTIFACT }}  
-Dsonar.organization=miw-upm-github  
-Dsonar.host.url=https://sonarcloud.io  
-Dsonar.token=\${{ secrets.SONAR\_TOKEN }}

**env:**

**GITHUB\_TOKEN:** \${{ secrets.GITHUB\_TOKEN }}

# Análisis Estático

## Sonarcloud. POM

```
<!-- coverage report -->
<jacoco.version>0.8.13</jacoco.version>
<!-- Official Plugin Sonar -->
<sonar-maven-plugin-version>5.1.0.4751</sonar-maven-plugin-version>

<!-- sonar plugin -->
<plugin>
  <groupId>org.sonarsource.scanner.maven</groupId>
  <artifactId>sonar-maven-plugin</artifactId>
  <version>${sonar-maven-plugin-version}</version>
</plugin>
<!-- coverage report -->
<plugin>
  <groupId>org.jacoco</groupId>
  <artifactId>jacoco-maven-plugin</artifactId>
  <version>${jacoco.version}</version>
  <executions>
    <execution>
      <id>default-prepare-agent</id>
      <goals>
        <goal>prepare-agent</goal>
      </goals>
    </execution>
    <execution>
      <id>default-report</id>
      <phase>post-integration-test</phase>
      <goals>
        <goal>report</goal>
      </goals>
    </execution>
  </executions>
</plugin>
```

# Análisis Estático Sonarcloud

Máster en Ingeniería Web / es.upm.miw.iwvg-devops master +

Overview Issues Measures Code Activity Administration ▾

Quality Gate Passed

Reliability [Measures](#)

 0 A  
started 10 days ago

Bugs

Security [Measures](#)

 0 A  
Vulnerabilities

 2  
Security Hotspots

Maintainability [Measures](#)

 2h A  
Debt

 21  
Code Smells

Coverage [Measures](#)

 55.4%  
Coverage

 14  
Unit Tests

Duplications [Measures](#)

 10.3%  
Duplications

 2  
Duplicated Blocks

# Calidad del código



## QA in action

- Sonarcube

## Ejercicios

- Integrar QA con un proyecto

# ¿Qué es la Entrega Continua?

*Continuous Delivery*

Práctica del desarrollo del software para liberar código en ciclos cortos

El objetivo es que las mejoras del desarrollo sean visibles para el cliente lo antes posible y poder tener una retroalimentación.

**Despliegue Continuo**  
*(Continuous Deployment)* resulta cuando las entregas se despliegan automáticamente.

Servidores  
Físicos

Servidores  
Virtuales

La Nube

IaaS

Infraestructura  
como Servicio  
  
Se ofrece  
almacenamiento,  
redes y servidores

PaaS

Plataforma como  
Servicio  
  
Se ofrece un entorno  
para distribuir  
aplicaciones.

SaaS

Software como  
Servicio  
  
Se ofrecen  
aplicaciones

FaaS

Funciones como  
servicio serverless  
  
AWS Lambda,  
Google Cloud  
Functions

CaaS

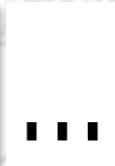
Contenedores como  
servicio  
  
AWS ECS EKS ,  
Google Kubernetes  
Engine

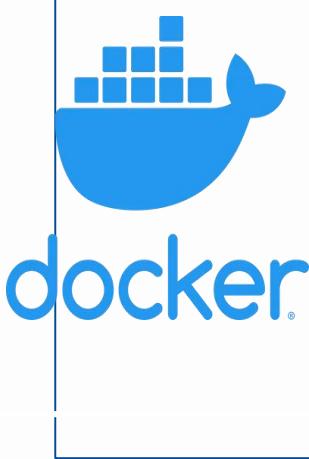
DBaaS

Bases de datos  
como servicio  
  
AWS RDS, Firebase,  
MongoDB Atlas

StaaS

Almacenamiento  
como servicio  
  
AWS S3, Google  
Cloud Storage,  
OneDrive





## Docker

- Docker es un proyecto de código abierto que automatiza el **empaquetamiento, distribución y ejecución** de aplicaciones dentro de contenedores de software.
- Es una plataforma que simplifica el desarrollo y despliegue de aplicaciones al hacerlas portables, eficientes y escalables.
- Es una herramienta clave en *DevOps*.
- <https://www.docker.com/>

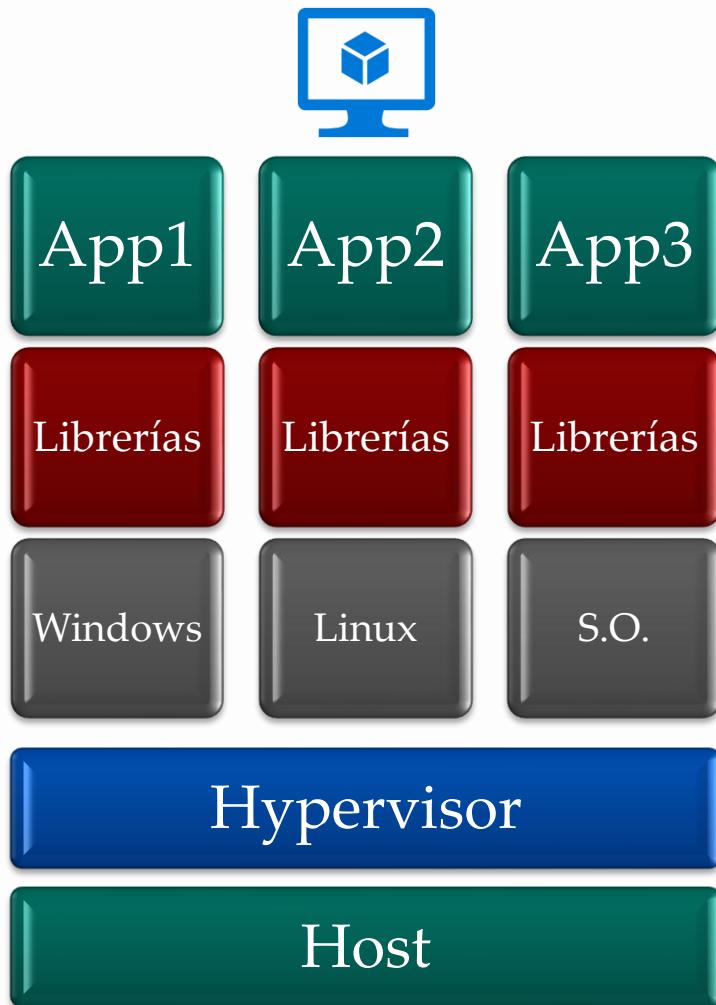


## Docker Hub

- Repositorio de imágenes de Docker
- <https://hub.docker.com/>

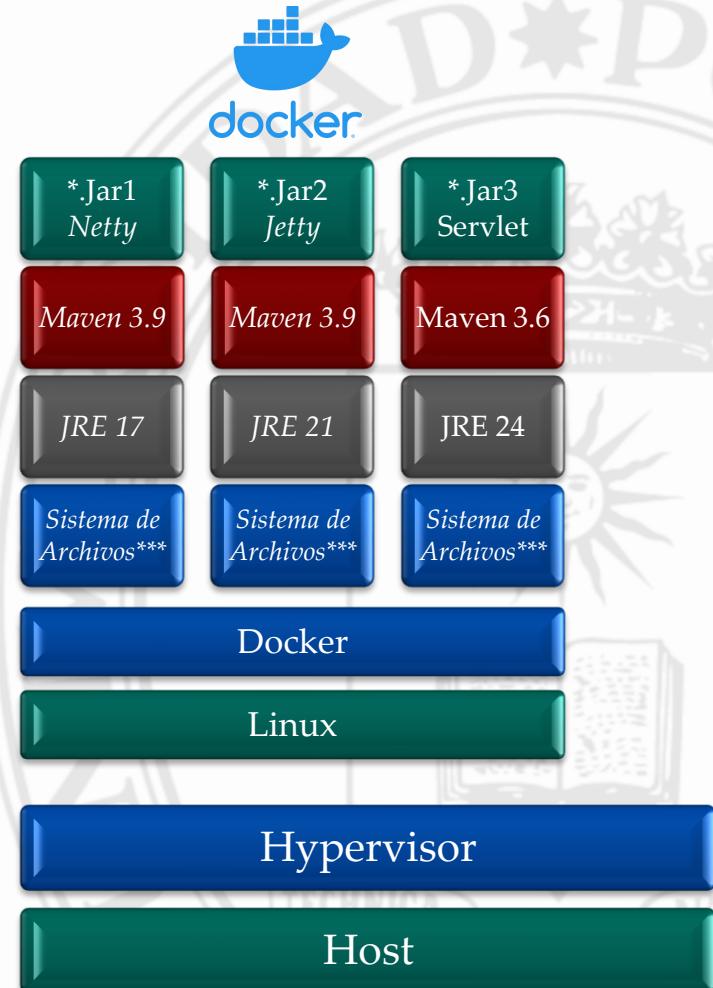
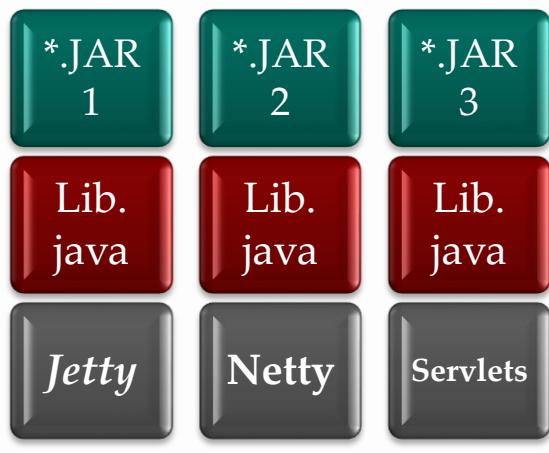
# BETCA

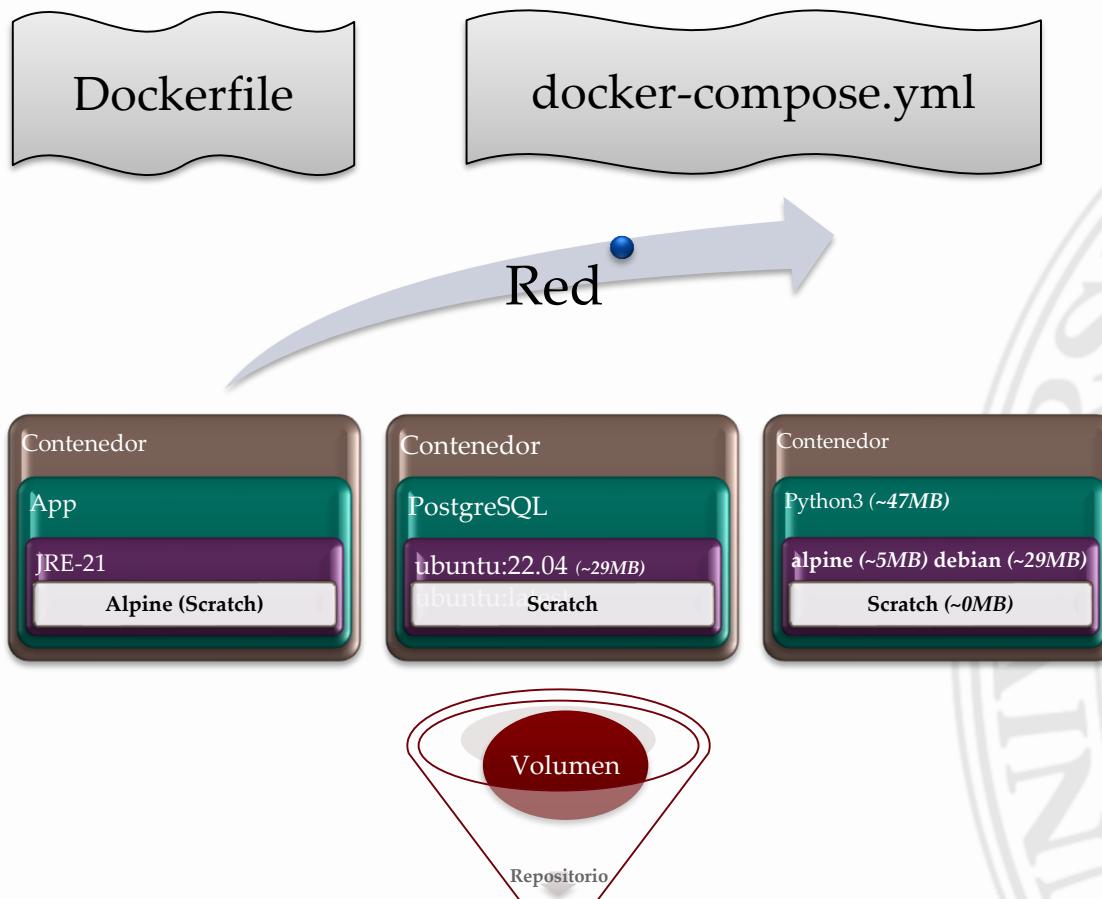
## Host VS Virtual Machine & WAR



# BETCA

## Jar & Docker





Docker

## Docker

- **Imagen.** Es una plantilla inmutable que contiene lo necesario para ejecutar una aplicación (*código, dependencia, configuración...*). Se construyen a partir de un fichero *Dockerfile* de tipo texto.
- **Imagen scratch.** Es el origen
- **Contenedor.** Unidad de software empaquetado para su ejecución de una imagen. Puede contener *variables de entorno*. Se puede limitar el uso de memoria y CPU.
- **Red.** Docker pude crear redes para que los contenedores se puedan comunicar entre si.
- **Volumen.** Son almacenamientos persistentes de datos y sobreviven al ciclo de vida de los contenedores.
- **Docker Compose.** Es una herramienta para definir y ejecutar múltiples contenedores, se define mediante el archivo *docker-compose.yml*.
- **Kubernetes.** Es una herramienta para administrar los contenedores en producción.

## Instalación

- <https://www.docker.com/>
- <https://hub.docker.com/>

## Comandos

- |                                      |  |
|--------------------------------------|--|
| • docker comando --help              | # Muestra la ayuda del comando indicado  |
| • docker pull postgres[:15.10]       | # Se baja una imagen, si no se pone versión sería <i>latest</i>                    |
| • docker images                      | # Muestra las imágenes   |
| • docker image rm <id>               | # Borra una imagen   |
| • docker build -t <imagen:version> . | # Construye una imagen a partir de <i>Dockerfile --tag</i> , o -f filename         |
| • docker create --name<name> <image> | # Crea un contenedor a partir de una imagen  |
| • docker ps [-a]                     | # Muestra los contenedores --all   |
| • docker rm <id>                     | # Borra un contenedor  |
| • docker start -a -i <id>            | # Ejecuta un contenedor, -a --attach, -i:--interactive                             |
| • docker stop <id>                   | # Para un contenedor en ejecución  |
| • docker run -d -p 8080:8080 <image> | # Construye un contenedor y ejecuta a partir de imagen [-it] --interactive & --tty |
| • docker logs [-f] <id>              | # Muestra los logs de un contenedor  |
| • docker compose up                  | # Construye, (re)crea, inicia y adjunta a partir de <i>docker-compose.yml</i>      |
| • docker compose down                | # Detiene y elimina  |
| • docker volumen ls                  | # Lista los volúmenes existentes   |
| • docker network ls                  | # Lista las redes  |
| • ...                                |  |

#### Fichero: *Dockerfile*, en la raíz del proyecto

- Tipo txt

```
# Etapa1
FROM maven:3.9.9-eclipse-temurin-21 AS build
WORKDIR /app
COPY pom.xml .
RUN mvn dependency:go-offline -B
COPY src ./src
RUN mvn clean package -DskipTests

# Etapa2
FROM eclipse-temurin:21-jre-alpine
WORKDIR /app
COPY --from=build /app/target/*.jar app.jar
EXPOSE 8081
CMD ["java", "-jar", "app.jar"]
```

Parte de una imagen que combina maven(3.9.9) & JDK(21)

Establece el directorio de trabajo dentro del contenedor

Copia el *pom.xml* de la ruta raíz del proyecto a la carpeta de trabajo del contenedor

Ejecuta el comando maven para que se descargen las dependencias en el contenedor

Solo copia los fuentes del proyecto al contenedor

Ejecuta el comando maven para crear el \*.jar

Parte de una imagen JRE(21)

Establece el directorio de trabajo dentro del contenedor

Copia el \*.jar generado del contenedor de construcción

Habilita el puerto 8081 para que sea escuchado

Establece un comando para cuando se inicialice el contenedor: java –jar app.jar

# BETCA

## *docker-compose.yml*

```

version: '3.9'
services:
  database:
    image: postgres:15.10
    container_name: postgres-db
    restart: always
    environment:
      POSTGRES_USER: postgres
      POSTGRES_PASSWORD: postgres
      POSTGRES_DB: tpv
    ports:
      - "5432:5432"
    volumes:
      - pgdata:/var/lib/postgresql/data
    labels:
      - "es.upm.miw.betca-tpv-user=database"
  app:
    build: .
    container_name: java-app
    restart: always
    depends_on:
      - database
    ports:
      - "8081:8081"
    environment:
      SPRING_DATASOURCE_URL: jdbc:postgresql://database:5432/tpv
      SPRING_DATASOURCE_USERNAME: postgres
      SPRING_DATASOURCE_PASSWORD: postgres
      SPRING_JPA_HIBERNATE_DDL_AUTO: update
    labels:
      - "es.upm.miw.betca-tpv-user=app"
  volumes:
    pgdata:
      name: "postgres-data"

```

Versión de Docker utilizada

Contenedor

Indica que cuando el contenedor de error, o se reinicie el host, se arranque de nuevo

Facilita la organización y gestión de contenedores

Contenedor

Se construye a partir del fichero Dockerfile

Nos indica que hasta que no se inicie el contenedor de BD, este no puede arrancar

Puentea el puerto interno 8081 al puerto de expuesto del contenedor 8081

Se configura Spring

## GitHub. Packages Repository

### GitHub Packages Repository

- **GitHub Packages** es un servicio de alojamiento y gestión de *paquetes de software* que permite a los desarrolladores almacenar y compartir paquetes de manera privada o pública.
- Los paquetes se integran perfectamente con *APIs de GitHub*, *GitHub Actions* y *webhooks* (notificaciones automáticas entre sistemas por HTTP) para crear un flujo de trabajo *DevOps* completo.
- **GitHub Packages** soporta varios registros de paquetes comunes, como *npm*, *RubyGems*, *Apache Maven*, *Gradle*, *Docker* y *NuGet* (.NET).

# BETCA

## Packages Repository

The screenshot shows a GitHub Packages repository page for the user `miw-upm`. The repository is named `betca-tpv-user` and has a version `4.5.19-Release`. The page includes sections for installing from the command line (using Docker), recent tagged image versions, and a download count of 3 for the latest version.

**Recent tagged image versions**

Version	Published	Downloads
<code>latest</code>	About 5 hours ago	3
<code>4.5.19-Release</code>	About 5 hours ago	3
<code>4.5.18-Release</code>	About 6 hours ago	2
<code>4.5.17-Release</code>	About 6 hours ago	2
<code>4.5.16-Release</code>	About 6 hours ago	2

# BETCA

## Despliegue Continuo

```
name: CD
on:
  push:
    branches: ['master']

env:
  REPOSITORY: ${{ github.repository }} # <user>/<repo>
  GHCR_URL: ghcr.io # GitHub Container Registry

jobs:
  cd:
    name: Build & Push GitHub Package & Deploy on Render Docker Image
    runs-on: ubuntu-22.04
    steps:
      - name: ⚡ Checkout repository
        uses: actions/checkout@v4
        with:
          fetch-depth: 0

      - name: 🐾 Set up Java
        uses: actions/setup-java@v4
        with:
          distribution: temurin
          java-version: 21
          cache: maven

      - name: 📁 Extract artifact from pom
        run: |
          echo "VERSION=$(mvn help:evaluate -Dexpression=project.version -q -DforceStdout)" >> $GITHUB_ENV
          echo "ARTIFACT=$(mvn help:evaluate -Dexpression=project.artifactId -q -DforceStdout)" >> $GITHUB_ENV
```

Fichero *yaml* para el *Despliegue Continuo* disparado por la rama *master*

Se generan dos variables de entorno: **REPOSITORY**: *miw-upm/betca-tpv-user* & **GHCR\_URL**: *ghcr.io*

Se generan dos variables de entorno: **VERSION**: *1.1.2-Release* & **ARTIFACT**: *betca-tpv-user*

# BETCA

## Despliegue Continuo

- name: 🛡 Login to GitHub Container Registry

uses: docker/login-action@v3

with:

registry: \${{ env.GHCR\_URL }}

username: \${{ github.actor }}

password: \${{ secrets.GHCR\_PERSONAL\_ACCESS\_TOKEN }}

1. Se crea un token de acceso personal en el perfil de la cuenta de GitHub con los permisos: *write:packages & read:packages delete:packages & repo*.
2. Se crea una variable de entorno en el repositorio con el token: *GHCR\_PERSONAL\_ACCESS\_TOKEN*

- name: 🏰 Build & Push Docker image

uses: docker/build-push-action@v4

with:

context: .

push: true

tags: |

\${{ env.GHCR\_URL }}/{{ env.REPOSITORY }}:\${{ env.VERSION }}

\${{ env.GHCR\_URL }}/{{ env.REPOSITORY }}:latest

Se construye a partir del fichero Dockerfile que esta en la raíz del proyecto

Se sube las dos referencias, la versión creada y la etiquetada como latest

Se crean dos etiquetas

# BETCA

## Despliegue Continuo

Fichero *yaml* para el *Despliegue Continuo* disparado por la rama *master*

Si la petición es incorrecta, nos devuelve un error. Pero no se controla si la app se levanta adecuadamente

Se llama a un end-point del APIRest de **Render**, para indicarle que se realice un nuevo despliegue

```
- name: 🚀 Deploy on Render
run: |
  curl --fail -X POST "https://api.render.com/deploy/${{ secrets.RENDER_DEPLOY_HOOK_TOKEN }}" \
  && echo "$input Deployment succeeded" \
  || (echo "❌ Deployment failed" && exit 1)
```

Si algo sale mal en la llamada, da por fallido esta etapa del despliegue

1. En **Render**, en la *Settings* del repositorio, se debe mirar el *Deploy Hook* y copiar el *token* de acceso.
2. Crear en el repositorio, en *Secrets and variable*, en *Actions*, un *new repository secret* con el token

# BETCA

Render. <https://render.com/>



Soporta Despliegue Continuo:  
*GitHub, GitLab & Bitbucket*

Se pueden crear distintos tipos de servicios: *Web estática, Servicio Web, DataBase (PostgreSQL, MySQL, MongoDB)...*

Soporta varios lenguajes: *Node.js, Python, Ruby, Go... (Java) & Docker image*

# BETCA

## Render

Hobby	Professional	Organization	Enterprise
For hobbyists, students, and indie hackers.	For small teams and early-stage startups.	For larger teams with complex needs.	For ultimate power and customization.
<b>\$0 USD</b> per month	<b>\$19 USD</b> per user/month	<b>\$29 USD</b> per user/month	<b>Custom pricing</b>
<a href="#">+ Compute Costs</a>	<a href="#">+ Compute Costs</a>	<a href="#">+ Compute Costs</a>	<a href="#">+ Compute Costs</a>
<a href="#">Get Started</a>	<a href="#">Get Started</a>	<a href="#">Get Started</a>	<a href="#">Contact Sales</a>

**Blueprints** are Render's infrastructure-as-code (IaC) model for defining, deploying, and managing multiple resources with a single YAML file:



# BETCA

## New Web Service

Source Code

Name (*único*)

Region  
*Frankfurt (EU Central)*

Instance Type  
*Free \$0 / month*

Environment Variables

Health Check Path  
*/actuator/health*

### Source Code

Git Provider	Public Git Repository	<a href="#">Existing Image</a>
--------------	-----------------------	--------------------------------

### Image URL

Deploy an image from a Docker registry

[ghcr.io/miw-upm/betca-tpv-user:latest](#)

### Credential (Optional)

[RENDER\\_DEPLOY\\_TOKEN](#)

[Connect →](#)

### Add Credential

Learn more about storing external [registry credentials](#).

Name

Registry

[GitHub](#)

You'll need to create a personal access token with the required `read:packages` permission in GitHub [Settings > Developer Settings](#).

Username

Personal Access Token

[Cancel](#)

[Add Credential](#)

### WEB SERVICE

**betca-tpv-user:latest**

[Image](#)

[Free](#)

[Upgrade your instance →](#)

[miw-upm / betca-tpv-user](#) [latest](#)

<https://betca-tpv-user-latest.onrender.com>

### Environment Variables

Set environment-specific config and secrets (such as API keys), then read those values from your code. [Learn more](#).

 **Render**

Key	Value
DATABASE_PASSWORD	tbu[REDACTED]Toi2f
DATABASE_URL	jdbc:postgresql://dp[REDACTED]80-a:5432/tpv_user
DATABASE_USERNAME	miw_upm
JWT_SECRET	791e[REDACTED]ec3e0
PORT	10000

